Increasing Customization in a Location-Based Augmented Reality Software Platform

by

Victoria Christine Jones

B.S., Massachusetts Institute of Technology (2015)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2017

Signature redacted

Author .....
Department of Electrical Engineering and Computer Science
August 10, 2017

Signature redacted

Certified by..
Eric Klopfer
Professor, MIT Scheller Teacher Education Program (STEP)
Thesis Supervisor

Signature redacted

Accepted by .....
Christopher Terman
Chairman, Department Committee on Graduate Theses

Increasing Customization in a Location-Based Augmented Reality Software Platform

by

Victoria Christine Jones

Submitted to the Department of Electrical Engineering and Computer Science
on August 10, 2017, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

In this thesis, I designed and implemented a framework for adding customizable icons to an existing location-based augmented reality software platform. This platform is comprised of a browser-based editor and a mobile application. In the editor, my work included increasing the size and variety of the existing icon palette, as well as, including a new option for uploading custom images for icons. This framework allows for greater understanding during gameplay, which in turn, improves the user experience while using the mobile application. Additionally, I redesigned the user interface used for interacting with these icons in the editor. This UI redesign increases learnability and usability by creating parallels between the editor and the mobile application. A series of user testing sessions were conducted to determine the effectiveness of these new features.

Thesis Supervisor: Eric Klopfer
Title: Professor, MIT Scheller Teacher Education Program (STEP)

Thesis Supervisor: Lisa Stump
Title: Lead Software Developer, MIT STEP

Acknowledgments

This work would not have been possible without the emotional support of my family and friends. I am thankful to my thesis supervisors, Lisa Stump and Judy Perry, and my faculty advisor Eric Klopfer.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The MIT Scheller Teacher Education Program (STEP) works to bring "playfulness and engagement to the forefront of the teaching and learning process"[2]. Using this mentality, MIT STEP has developed several tools to enhance learning, one of which is TaleBlazer. TaleBlazer is a platform that allows users to make and play location-based mobile augmented reality (AR) games. Augmented reality games, such as TaleBlazer, allow users to interact with the real world and learn using smartphones, which have become commonplace. Users can connect the game they are playing to the world around them, which creates a more hands-on learning experience.

TaleBlazer users are broken into two different groups: game designers and players. These groups are not mutually exclusive as designers are also players. TaleBlazer games are created by game designers over a specified region in the world. These games usually tell a story about the location within the game area. Then, the game is played when the application uses a cell phone or tablet's built-in GPS to pinpoint a TaleBlazer player's location. Within a game, players move to interact with agents, which are virtual characters or objects. Aside from using the mobile application to play games, designers can create, edit, and save their own games by means of the TaleBlazer editor. With an online editor that utilizes a blocks-based programming environment, TaleBlazer enables beginner and expert programmers to develop their own unique creations about any topic of their choosing.

TaleBlazer is designed to reach children (or other players) who are interacting with the application for a short amount of time, such as on a field trip to a zoo or museum. While games can be located inside or outside, most TaleBlazer games are played outside, where there is a need for high contrast on the screen in order to be able to interact with the game.

Chapter 2

Motivation

Many mobile applications allow users to customize small aspects of the app in order to make the experience more personalized and usable for each user. The customizable features are app-dependent and help the user more easily achieve the application's goal. TaleBlazer game designers are asking for more ways to customize their games. These customizations will allow games to have more context and a greater narrative.

For example, TaleBlazer game designers can configure buttons to appear in their games. Game designers can change the words on these buttons, but icons also appear on the buttons. These button icons are static and depend on the type of action that the button performs. Thus, to a player, the meaning behind a button icon isn't immediately clear.

Customization features should be added to TaleBlazer with the purpose of giving the game designer more freedom to create greater theme and tone in each game. Customization also improves the user experience, since more information can be conveyed visually. In order to learn which customization features are most desirable, it is useful to look at other popular applications and see how they use customization to add meaning and ease of use to their product.

2.1 Popular Mobile Applications with Customizable Aspects

By researching the most popular mobile applications, I gained insight into how to use the real estate on the screen to convey information to the user without being overwhelming or confusing.

According to Quartz (qz.com), the top mobile applications in the US as of June 2015 were as follows [3] :

1. Facebook
2. YouTube
3. Facebook Messenger
4. Google Search
5. Google Play
6. Google Maps
7. Pandora Radio
8. Gmail
9. Instagram
10. iTunes Radio/iCloud

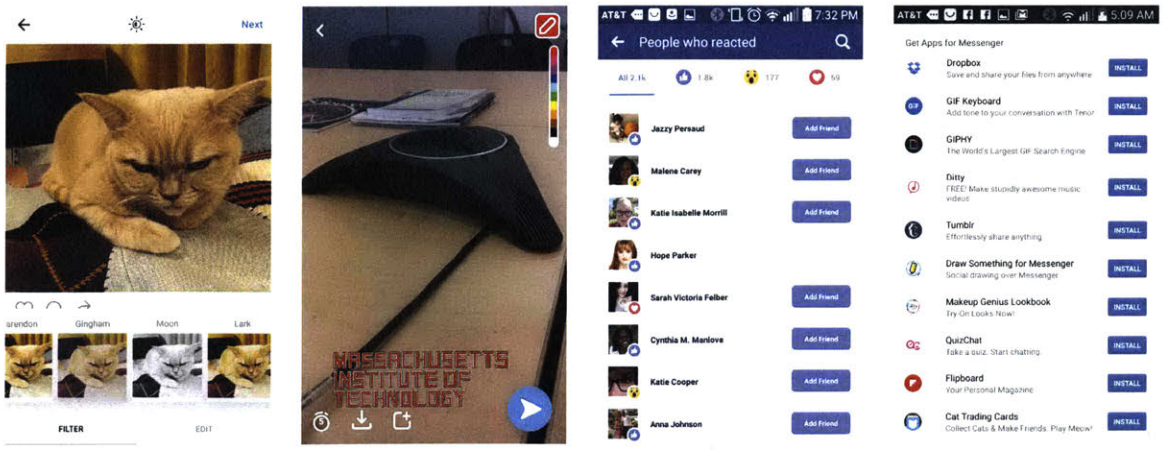
While none of these are augmented reality games, most, if not all, of these applications allow their users to customize some aspect of the application.

For example, Instagram allows users to customize the images that they are about to upload by adding filters. Since Instagram is a photo sharing application, it makes sense that users should be able to modify a photograph before uploading. Similar to Instagram, Snapchat is a photo sharing service that allows people to send images and videos to their friends. The images can be edited to include words, drawings, stickers, filters, or emojis before being sent. The main difference between Snapchat and Instagram is that Snapchat is designed so that most images or videos are temporary and deleted after 3-10 seconds. Recently, Snapchat users can save their photos or

videos to their "Memories" for personal viewing and there is a new "infinity" option for sending photos. The "infinity" option lets a person view an image for as long as they like and the photo is then deleted after they close the page.

Facebook is a social media application that connects people to their friends and family. Recently, instead of just having the "Like" button, Facebook has included a reaction panel that allows users to react to content like posts, articles, and videos with options such as "Like", "Love", "Haha", "Shocked", "Sad", and "Angry". This gives the user a wider array of emotions to choose from so that their proper feelings can be expressed. Users can also add feelings to posts so that their friends can tell how they are feeling at that moment. Facebook Messenger separates direct messaging from the main Facebook app. In these direct messages, users can customize their conversations in many ways. Users can create, delete, and edit groups. They can change the default color of the group, change the name of the group, and add or block members. Nicknames can be added to users in the group and the default icon can be adjusted. This allows each conversation to be unique or personalized to the group's content.

See Figure 2-1 for screenshots of all the aforementioned applications.



(a) Instagram (b) Snapchat (c) Facebook (d) Messenger

Figure 2-1: Screenshots of (from left to right) Instagram, Snapchat, Facebook, and Facebook Messenger illustrating the image customization capabilities in (a) Instagram and (b) Snapchat, a reactions page on (c) Facebook, and the downloadable add-ons available for customizing (d) Facebook Messenger.

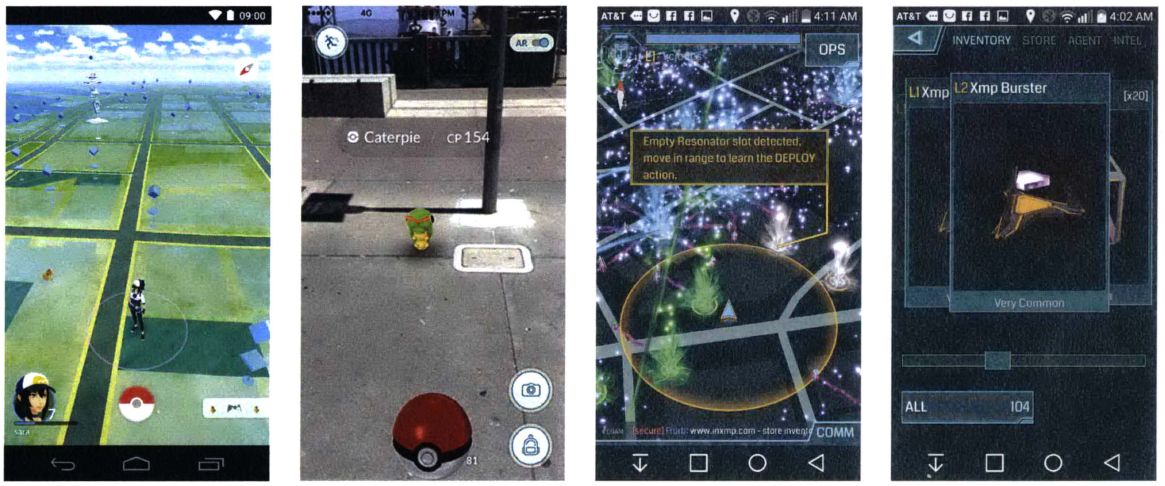
2.2 Other Existing Mobile AR Games

TaleBlazer is not the only location-based AR game available for mobile devices. Other games of this type exist with well-implemented (or poorly implemented) features. Therefore, if features are available in other applications that improve ease of use and understanding, it would benefit TaleBlazer to also incorporate similar features or capabilities.

There are a number of existing mobile augmented reality games that are location based. A popular game that recently launched this past year, Pokémon Go, is a prime example of such a game. In Pokémon Go, players explore the world around them to find and capture virtual Pokémon at real world locations. Important locations in the real world (churches, parks, public artworks, landmarks, etc.) are often pokéstops in the game. Players visit these pokéstops to collect items and find Pokémon. Pokémon Go's creator company, Niantic Inc., is also behind another augmented reality game, Ingress. Ingress is a science fiction game where players choose between two sides and battle for control of portals. The same pokéstops in Pokémon Go are called portals

in Ingress. The look and feel between the two games is altered because they have different target audiences: Pokémon Go is brighter and less cluttered because it has a wide target audience ranging from children to adults, while Ingress' target audience is mostly adults. Children and adults enjoy the simple layout of Pokémon Go while adults don't mind the intricacies of Ingress.

Ingress and Pokémon Go both utilize a dynamic map which allows the user to see which direction they are facing. Pokémon Go is less cluttered than Ingress because there are fewer features and screens needed for gameplay. Both games have advantages in terms of design and aesthetic that could be utilized to make TaleBlazer more visually appealing to users, but neither game allows much in terms of customization. In the beginning of each game, users can set a few choices (customize an avatar in Pokémon Go and pick a faction in Ingress), but aside from that, the games are pretty static.



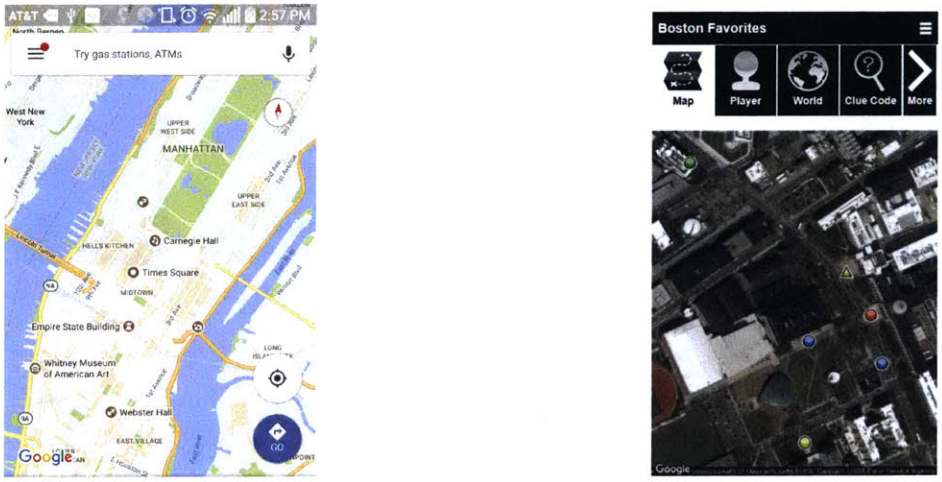
(a) Pokémon Go Map (b) Pokémon Go Capture Screen (c) Ingress Map (d) Ingress Gameplay

Figure 2-2: Screenshots of the (from left to right) Pokémon Go Map, Pokémon Go Pokémon Capture Screen, Ingress Map, Ingress Inventory screen

As you can see in Figure 2-2, both games are pretty crisp and modern. Pokémon Go is more minimal and bright in terms of the map, while Ingress has a darker

theme and visually looks more crowded. Both of these games have map icons that have meaning. In Figure 2-2, the strange blue icons on the Pokémon Go map are the pokéstops and the green/white/blue shiny icons on the Ingress map are the portals. On the Pokémon map in 2-2a, there is also a Pokémon visible that the player can walk to in order to try to capture that Pokémon. In the background of the Pokémon map, there is another special icon which is a Pokémon gym. Therefore, in Ingress and Pokémon Go, the player can see the icons and recognize exactly what will happen when they approach them.

In Figure 2-3, there are images of Google Maps and TaleBlazer on a mobile device. In the Google Maps image, there are different icons that are used to portray different kinds of location in New York City. For example, Carnegie Hall's icon contain a music note, which is intuitive because Carnegie Hall is a concert venue. The Whitney Museum of American Art has a building icon with an "M" in the center, which is intuitive because a museum starts with "M" and is a building. TaleBlazer, on the other hand, has unintuitive icons. The red circle in Figure 2-3 is placed over MIT, a private research university. The dark green circle is placed over Flour Bakery, a café in Cambridge. Unlike Pokémon Go, Ingress, and Google Maps, TaleBlazer does not have informative icons on their map screen.



(a) Google Maps (b) TaleBlazer

Figure 2-3: Google Maps (Left) and TaleBlazer map screen (Right) on mobile devices

2.3 Takeaways

Customization adds meaning and/or ease of use to applications. Instagram, Snapchat, and Facebook allow you to enhance the information that you want to display. Facebook's reaction feature allows users to choose from a palette of emotions instead of its previous implementation that allowed users to toggle between "Like" and "Neutral". This feature adds depth to the posts on their platform. Instagram and Snapchat allow users to personalize images so that they are perceived a certain way, thus, adding greater meaning to the images. Facebook Messenger gives users the ability to personalize each individual group such that they are easy to identify. Applications allow customization so that the application itself is personalized for each individual. This lets users to interact with the app in their own way. Personalizing an application allows the app to be used most efficiently by each user. Icons are important on mobile applications because they portray subtle meaning in a small amount of space. And space is incredibly important on small screens.

While the mobile application isn't customizable for the game player, the game designer should be able to use customization to add meaning and ease of use to their TaleBlazer games. The game creator should be allowed to enhance the game for better information portrayal and thus, a better user experience. And by allowing the game creator to customize their games, the game player is able to more easily play and understand the game. Since TaleBlazer is designed to reach players who are interacting with the application for a short amount of time, customizable features that improve their understanding of the game can result in a more visually engaging encounter with TaleBlazer. Players would be able to have a more positive experience within the desired play time.

For a TaleBlazer game designer, there is currently a challenge to create a themed game. For example, a game about pirates will look very similar to a game about space. Designers can change the story, images, and location within a TaleBlazer game, but

the overall look and feel does not vary. Each TaleBlazer game utilizes the same symbols, so it is virtually impossible to create themes, such as those seen in Ingress and Pokémon Go. Additionally, players don't have time to comprehend TaleBlazer's abstract icons that are used on the map and action buttons. Thus, increasing the number of customizable features (while also improving the existing ones) would give the game designer the option to produce these different themes.

Chapter 3

Previous Problems

The TaleBlazer editor could be improved to allow for greater customizable features for the game designer. For my project, I focused on adding customizable icons to the map and to the action buttons. Adding customization to these two features will have the biggest impact on the designer and the player since both groups will interact with the map view and most likely the action buttons. Customizable icons involve two parts: creating a more diverse built-in palette of icons and including the option to upload custom images for icons.

Map icons and action button icons are aspects of the game that TaleBlazer game designers would like to customize. TaleBlazer's actions buttons have preselected icons, one of which has a non-intuitive meaning. Thus, to a player, an action button icon can be confusing or unhelpful. Additionally, agent map icons are limited to 40 options: 4 shapes and 10 colors. These ambiguous symbols don't tell the player any information about the agent.

A more diverse palette and an option to upload custom images for icons will give the game designer the customizable options they desire. Consequently, a similar framework should be used for the map icons and the action buttons since they have the same criteria for their icons.

3.1 Customizable Agent Map Icons

In TaleBlazer, an agent has a map icon that represents that agent on the map. Previously, the agent map icons were limited to 4 shapes (circles, squares, crosses, and triangles) and 10 colors. So there are only 40 possibilities for agent map icons. However, it would be more impactful to include a more diverse palette and an "Upload Icon" button for agents. This way, the game designer has the option to use one of the defaults or upload their own images. The default palette should be expanded from the original 40 options to include a more diverse palette of images. This way, game designers could avoid uploading images that could produce style irregularities between map icons in their game. Figures 3-1 and 3-2 illustrate the previous options for agent icons. In Figure 3-1, the map icon selected for this particular agent is a red circle.

Figure 3-1: The Agent Tab in the Old TaleBlazer Editor (background) with the "Pick Icon" Screen (foreground)

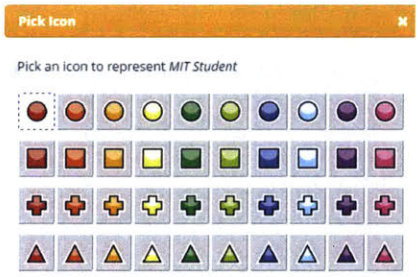


Figure 3-2: The "Pick Icon" Screen in the TaleBlazer Editor (Old Version)

In Figure 3-3, the map icon is shown on the map tab, which represents where this agent is in the world.

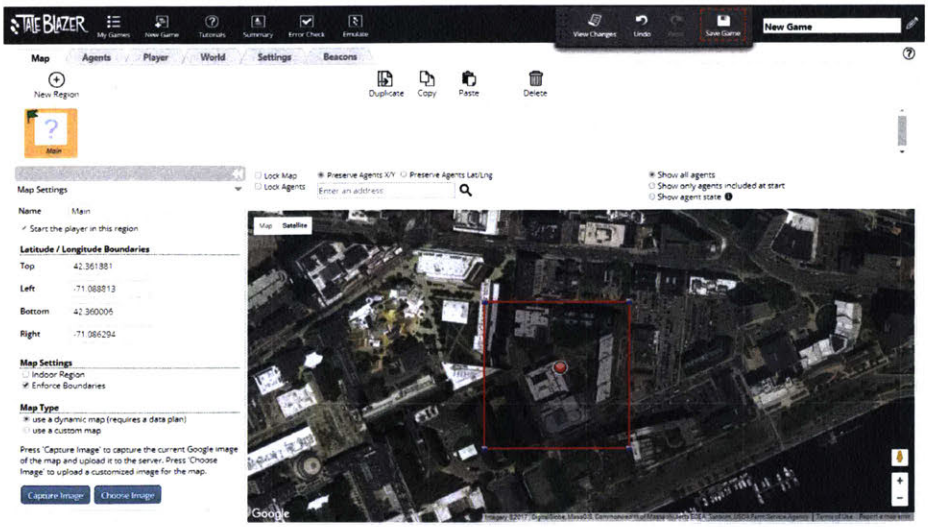


Figure 3-3: The Map Tab in the TaleBlazer Editor (Old Version) with the red circle representing an agent

Currently, TaleBlazer game designers can customize the images that are associated with agents that are used throughout the game. Figure 3-4 shows an instance of the image picker that is used to select these images. Four different file extensions are supported for agent images: jpg, png, jpeg, and gif. For the customizable agent map icon feature, I used a similar menu to the one shown in Figure 3-4 for customizable agent images. At the moment, gifs are not supported for agent icons (but this could be an area for future work).

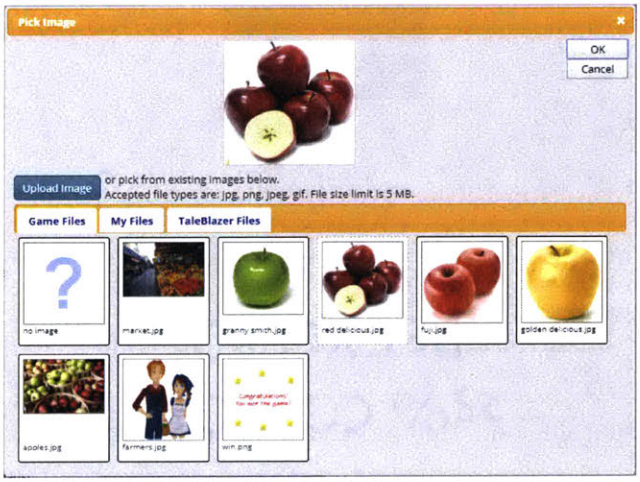


Figure 3-4: An example of the Image Picker in TaleBlazer Editor with some user uploaded images

3.2 Customizable Action Button Icons

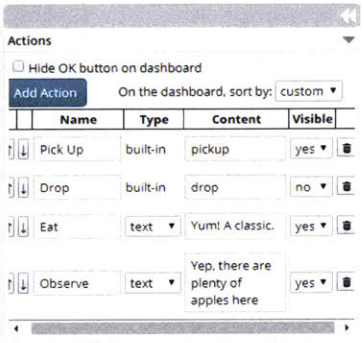
In TaleBlazer, entities like agents and roles can have action buttons. There are four types of action buttons in TaleBlazer: text, video, script, and built-in. Text buttons display text when they are pressed. Video buttons play videos when they are pressed. Script buttons execute code when they are pressed. The three built in buttons ("Pick Up", "Drop", and "Give") are used to support TaleBlazer's inventory system. Previously, each action button had its own icon depending on the type of action. As shown in Figure 3-5, text actions had a speech bubble icon and script actions had a star as the icon. There was also a play button icon for video actions. The video and speech bubble icons may be intuitive, but the star icon is not informative to a player at all (there is no meaning behind why that icon is a star). Also, the speech bubble icon can be more informative depending on the information that the designer is trying to convey.

Figure 3-5: Action Buttons on TaleBlazer Mobile (Old Version)

Therefore, it's useful to include a default palette of icons and an "Upload Icon" button for action buttons. This way, the game designer can choose to use the default icons, or upload a custom icon that is more aligned with what the button actually does. For example, in Figure 3-5, the "Open Chest" button can now have a hand icon instead of a star and the "Find Treasure" button can have a pile of treasure instead of the speech bubbles.

In Figure 3-5, the colorful treasure chest image is called the agent image. The entire screen in Figure 3-5 is called the agent dashboard. The game designer can upload their own custom images for the agent image using the screen shown in Figure 3-4 on page 27.

Formerly, game designers had trouble adding buttons to agents in the previous version of TaleBlazer. As shown in Figure 3-6a, the "Actions" table is used to create actions that will become buttons on the mobile application (Figure 3-6b).



(a) "Actions" Table in TaleBlazer Editor (Old Version)

(b) Corresponding buttons on TaleBlazer Mobile (Old Version)

Figure 3-6: The old version (left) of the "Actions" Table in the TaleBlazer Editor and the corresponding buttons on the Mobile (right)

Chapter 4

Solution - A Framework for Customizing Icons

Customizable icons give the game designer the freedom to add meaning and ease of use to their TaleBlazer game. Thus, I found a palette of potential icons that can be used on action buttons and the map (and potentially, on the tabs in the future). An icon palette makes the game appear more cohesive while also allowing for a diverse set of icon options. A diverse icon palette also helps the designer avoid potential problems with uploading custom images. For example, uploading images can be time consuming for the designer. The size of the image is important since the image needs to look good on a number of different screen resolutions. Also, mismatched icon images can ruin the aesthetics of the game and make it harder to play and understand.

Thus, the diverse icon palette should be useful for most game designers and advanced designers can use the custom upload option, if they so desire.

4.1 The Icon Palette and Uploading Custom Icons

When dealing with custom icons, TaleBlazer should have two components: a more diverse icon palette and an option to upload custom images for icons. Thus, I created a new framework for the map icons and the action button icons to use since they

have the same icon criteria. This common framework makes icons simpler for the TaleBlazer team to understand.

The action buttons previously had predetermined icons and the map icons previously stored an icon shape and an icon color. The new code framework eliminated both of these old ideas so that both map icons and action button icons can be supported using the same code, which makes the code easier to maintain, understand, and enhance.

4.1.1 Icon Palette: Fonts vs. Images

When including an icon palette, there were two main options: a font or a collection of images. When deciding between which to use, I had to consider multiple criteria as outlined in Table 4.1. Most importantly, TaleBlazer needed a diverse icon palette that scaled on different screen resolutions. I also weighed easily expandable icon palettes more highly.

Criteria	Font(s)	Images
Diverse set of Images	Need to find the smallest number of fonts that gives the greatest diversity	Find all individual images
Adjustable size for different screen resolutions	True	Possible Pixilation
Easy to Expand	Needs to be in the font already or find new font	Find new images
Adjustable Color	True	False
Effect on Mobile App Size	Depends on font(s) size(s)	Depends on images sizes and number of images
Monochromatic[1]	True	True or False

Table 4.1: The Criteria for the Icon Palette

[1]Multicolor icons look good in most applications, but might be hard to see in mobile TaleBlazer games outside. Thus, a monochromatic palette is more favorable.

Fonts

There are many fonts available that include a sequence of images (for example, the wingding font). Fonts are useful because the color and size can easily be adjusted. This means that a font can be resized without being pixelated and the color can be easily changed. A valuable icon font would be large enough to include a plethora of useful images. Thus, the TaleBlazer team would be able to incorporate the icons we want to use now and easily add in new characters later (without having to find another font).

Fonts are monochromatic. An upside to this is that a potential future feature could be allowing the game designer to change the color of the font icon. A potential downside is that designer can't make the font icons multicolored. However, while multicolored icons might be more visually appealing, they might make a TaleBlazer game harder to play because multicolored icons are hard to see outside. Thus, using something monochromatic might be a plus.

One drawback with fonts is that the entire font needs to be included in the TaleBlazer mobile app (so that a specific icon can be used). So including a number of different fonts would increase the size of the mobile application. Whether the designer used a font icon or not, those fonts would have to be included. Another issue is that fonts can lack diversity or be very repetitive, so it's hard to find a font that has a wide variety of images that we can use.

Also, many fonts include icons that aren't relevant to TaleBlazer. These icons could be symbols or images that are too specific or too vague to be used by most game designers. Some fonts are repetitive and have many variations of the same symbol (For example, a full moon, a crescent moon, a half-moon etc.). A summary of the pros and cons of using a font for the icon palette is included in Table 4.2.

Font(s)	
Pros	Cons
Adjustable color and size	Entire font must be included in the mobile application (increases the size of the application)
Adjustable size without pixilation	Can be repetitive (contain variations of the same symbol)
	Can contain unusable/irrelevant symbols (like symbols for decoration)
	Monochromatic
	Limited to the icons within the font (need to add new fonts if you want to include more icons)

Table 4.2: The Pros and Cons of using Fonts for the Icon Palette

Images

Images can be general. As in, the TaleBlazer staff can easily find an icon of an image and upload it into the default palette. So, diversity is not a problem when it comes to using images for the icon palette. Using images for the icon palette is also expandable. It's easy to find and add a highly requested icon to the existing palette.

The problem with using individual images is that it can take a long time to find and upload each and every image the game designers want to use. Size is also important when it comes to using images for icons. If an icon image is too small, it will look bad on the map or button. Icon images on the map and buttons are enlarged to fit inside the assigned box. They are only compressed if they are too big to fit in the assigned box [2]. Therefore, if the icon image is smaller than the box, it might be pixelated when it is enlarged on the map or button. And if the icon image is bigger than the box, it might look bad when it is compressed. A summary of the pros and cons of using a group of images for the icon palette is included in Table 4.3.

[2]For buttons, the assigned box is 10% of the device's shortest dimension. For the map, the assigned box is 6% of the device's shortest dimension.

Images	
Pros	Cons
Monochromatic or Colorful	Need to pick a good size that won't be pixelated or too small on different screen resolutions
Can include more images without rebuilding the mobile application	Have to find each individual image

Table 4.3: The Pros and Cons of using Images for the Icon Palette

Choosing to use a Font

I ended up selecting a font for the icon palette. The biggest problem stopping me from using a font for the icon palette was finding a large enough font that contained useful icons for TaleBlazer. WebHostingHub Glyphs [11] is a free font available from Font Squirrel that contains 1,514 glyphs (symbols). Due to its size, this font contains a large number of usable icons which should make most game designers happy. Secondly, using a font for the icon palette ensures that the icon palette icons will look good on different screen resolutions. Also, since I found one large font, I only needed to include one new font in the mobile application. The font palette is included in the "Built In" Section of the Icon Picker I redesigned (Section 4.3.1 on page 47).

Criteria	Font(s)	WebHostingHub Glyphs
Diverse	Need to find the smallest number of fonts that gives the greatest diversity	1,514 glyphs
Adjustable size for different screen resolutions	True	True
Easy to Expand	Needs to be in the font already or find new font	Start with a few hundred and expand if needed
Adjustable Color	True	True
Effect on Mobile App Size	Depends on font(s) size(s)	One large font
Monochromatic[3]	True	True

Table 4.4: The WebHostingHub Glyphs Font meets most of the criteria

[3]Multicolor icons look good in most applications, but might be hard to see in mobile TaleBlazer games outside. Thus, a monochromatic palette is more favorable.

4.1.2 Custom Uploaded Icons: Storing Uploaded Icons on the Server

Most game designers will use the diverse font icon palette for their icons. However, some advanced users will want to upload their own custom icon images. Thus, I needed to modify the way images are stored in the server.

Previously, there was no framework for storing icons on the server and creating that framework produced a tricky situation for the game designer.

For custom agent images that appear on the agent dashboard (See Section 3.2 and Figure 3-4), the pre-existing upload code compressed images to 800x800 pixels and stored them on the TaleBlazer server[4]. This size was not appropriate for icons which are displayed on a smaller percentage of the mobile screen than agent images[5]. Thus, I modified the pre-existing upload code when storing icon images. Due to different resolutions on different screens, I decided to make the maximum dimension 250 pixels. This means that icons larger than 250x250 pixels will be resized with the aspect ratio preserved (and the largest dimension will be 250 pixels). My choice to store the images at this size was due to the size the icons are shown on mobile devices. Currently, the highest resolution device that can use TaleBlazer will show icons at a maximum size of 216x216 pixels. By saving the icon images at 250x250 pixels, the icons should never be compressed on the server and then enlarged when they are displayed on the mobile application (which could possibly cause pixelated icons).

Pixelated icons may still occur. If a game designer uploads a small icon, the image will be expanded to fit in the containing box when it's displayed on the mobile application (with the aspect ratio preserved). The designer has to pick an image that is: 1. Not too big that it's illegible when it's compressed and 2. Not too small that

[4]Meaning images larger than 800x800 were compressed to this size before they were stored on the server or images that are smaller than this size were saved as it

[5]For reference, the default agent map icons were 50x50 pixels previously.

it's pixilated when it's expanded. Thus, it's easier for a game designer to use the icons provided in the default palette (as they scale with the size of the screen).

Naturally, as higher resolution devices are added to the market in the future, the number 250 should be adjusted. In the future, TaleBlazer will need to store larger icon images. I would recommend that TaleBlazer adjust the number to 512x512 pixels in the future for two reasons: 1. There are many websites that have icons available at this size and 2. As mobile screen resolutions increase, larger images are needed to avoid pixilation.

4.2 Customizable Agent Map Icons

As aforementioned in Chapter 3.1 on page 26, a map icon represents an agent on the map. In the editor on the "Map" Tab, there is a preview of how all the agents will look. And on the mobile application under the "Map" Tab, the player can see the map icons (Figure 4-1).

(a) Old Version of "Map" Tab in TaleBlazer Editor

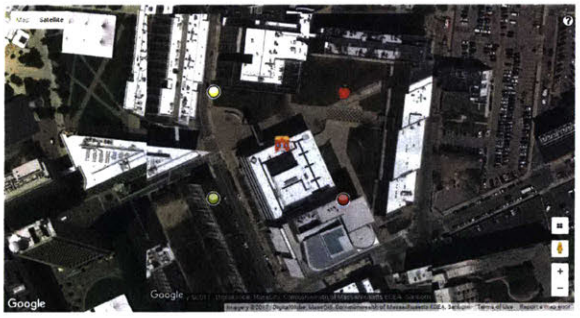
(b) Corresponding "Map" Tab in Mobile app

Figure 4-1: Map preview on the "Map" Tab in the old version of TaleBlazer (left) and the corresponding "Map" Tab on the mobile application (right)

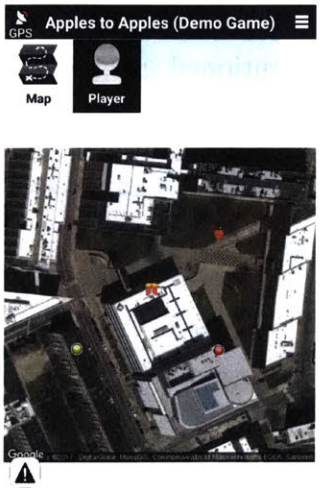
As mentioned in Chapter 4.1.1 on page 32, there is now a diverse icon palette that

is shared between the map icons and the action button icons. The palette can give more meaning to icons on the map and scale on different screen sizes.

Custom map icons are also supported in the revised TaleBlazer Editor. For example, if a user uploads a png file to represent an agent on the map, transparent background are supported. This means agent map icons can be shapes other than square or rectangular (As seen in the icons in Figure 4-2). The game designer can simply switch to the map tab in the TaleBlazer editor to see how their customized map icons will look on the map. This can let the designer know if the image they selected is too big[6] or if they accidentally selected an image without a transparent background. After reviewing, they can make the proper adjustments, if need be.



(a) New Version of "Map" Tab in TaleBlazer Editor



(b) Corresponding "Map" Tab in Mobile app

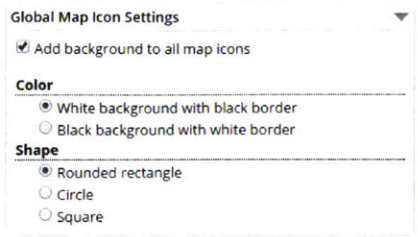
Figure 4-2: Map preview on the "Map" Tab in the revised version of TaleBlazer (left) and the corresponding "Map" Tab on the mobile application (right)

4.2.1 Global Map Icon Settings

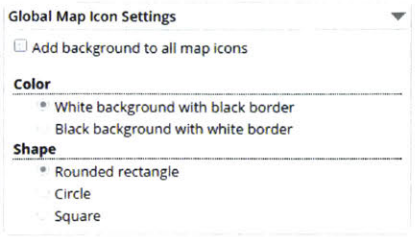
Visibility is an issue with map icons. Due to glare outside, low contrast, or the area on the map, an icon may be difficult to view, even if it is a good size. To deal with this problem, I added global map icon settings that allow the game designer to specify

[6]It will look bad on the map when it's compressed

that the map icon include either a black or white background with a border. This can be used to create contrast on the map. The game designer can use these backgrounds with the built-in icons or their own custom icons to make sure they are visible on the map. In case the game designer is not aware that this setting exists, I've included a descriptive hint in the dialog when uploading icons that will lead them to these new settings as shown in Figures 4-3 and 4-4.



(a) Map Icon Settings with a white rounded rectangle background and black border



(b) Map Icon Settings not used

Figure 4-3: New "Global Map Icon Settings" in Editor

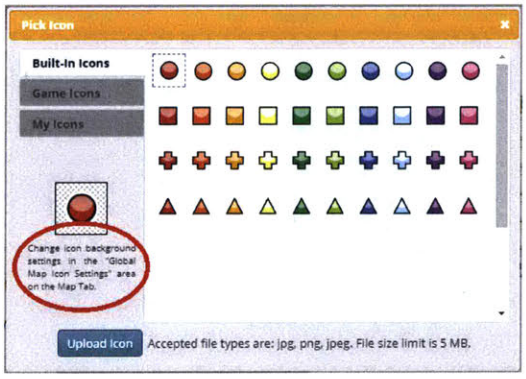
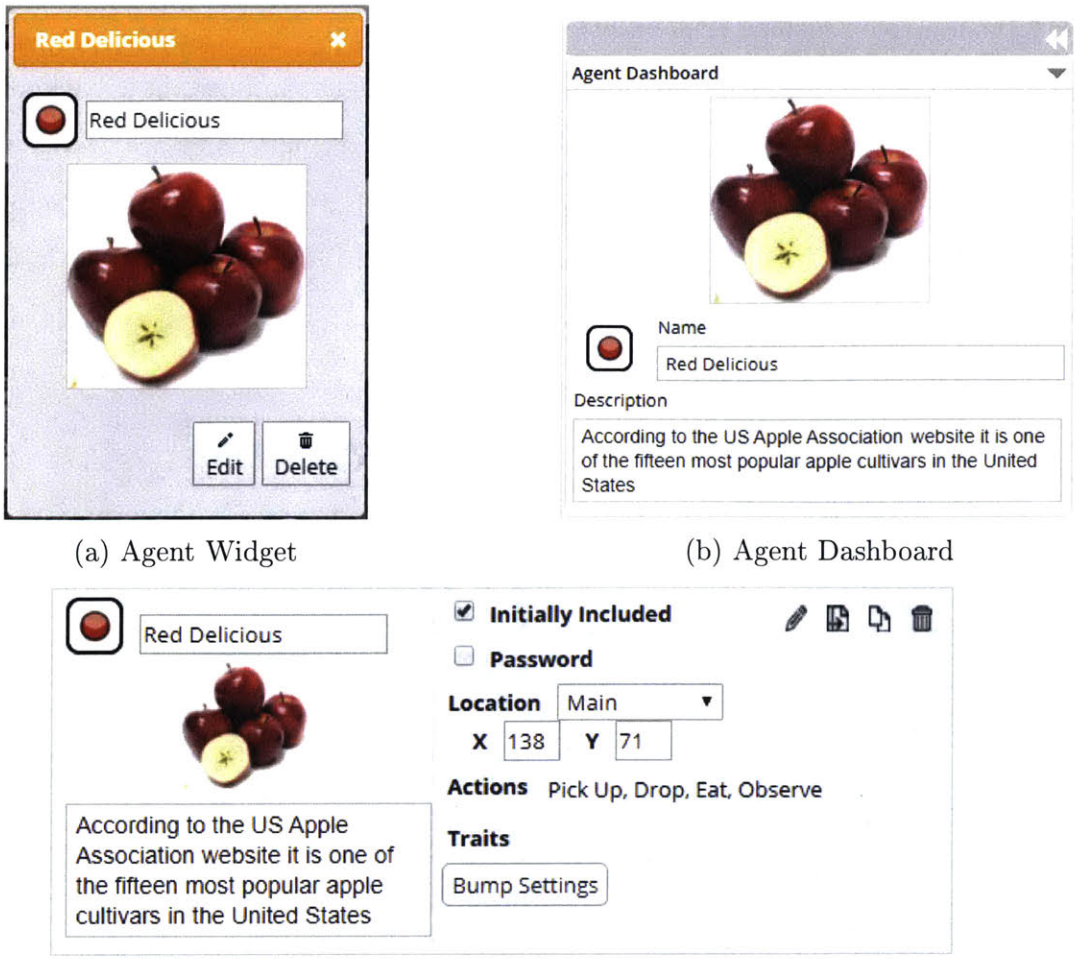


Figure 4-4: Help Dialog to help game designers find the new feature

As shown in Figure 4-5, when the game designer uses the map icon settings, those settings are reflected in the Agent Widget on the Map Tab, and the Agent Overview and Agent Detail View in the Agent Tab. These changes give the user more feedback about what icon background settings they are using on their agents.



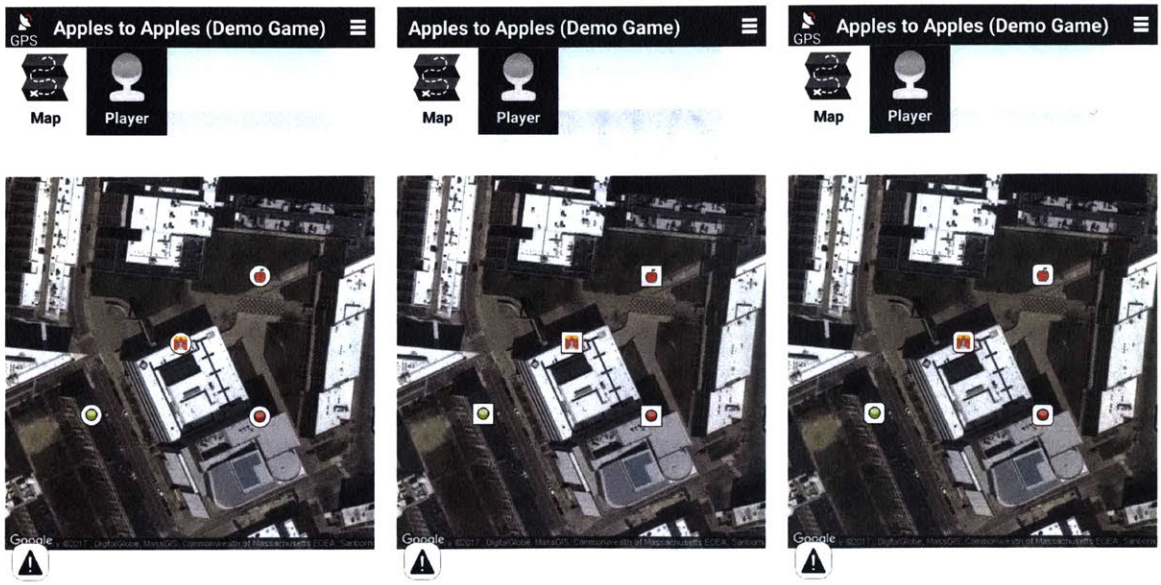
(a) Agent Widget

(b) Agent Dashboard

(c) Agent Overview

Figure 4-5: Changes in the New TaleBlazer Editor when the map icon settings are set to white rounded rectangle background with black border in the Agent Widget (Top Left), Agent Dashboard (Top Right) and Agent Overview (Bottom)

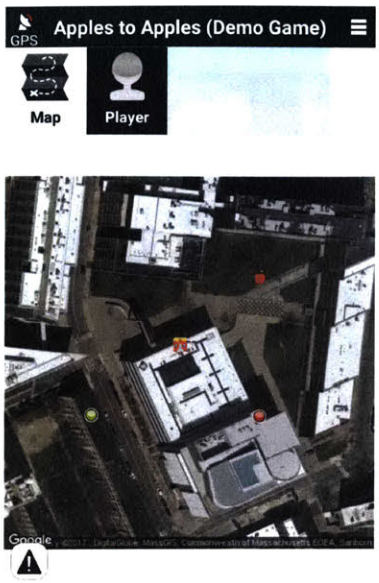
In Figure 4-6, I've included images of what the corresponding map icons look like on the mobile application using the different shapes.



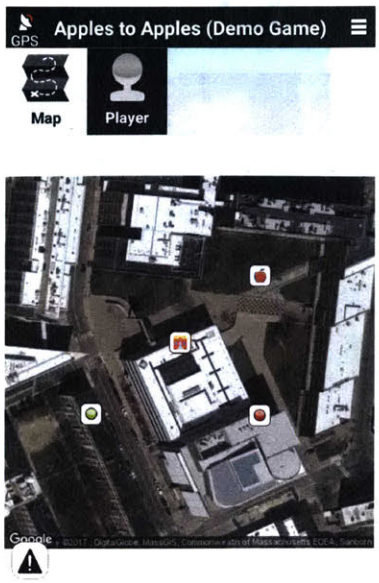
(a) Circle (b) Square (c) Rounded Rectangle

Figure 4-6: Images of the same mobile TaleBlazer game with a white background and black border. Shape set to Circle (left), Square (center) and Rounded Rectangle (right)

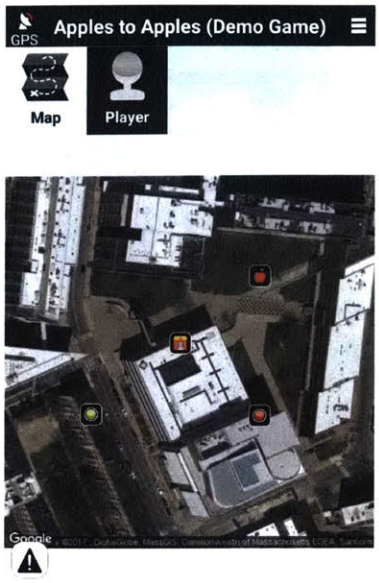
As I've previously mentioned, icons may be hard to view on different maps, so I've included some images of how various icons and backgrounds look on different maps (Figure 4-7).



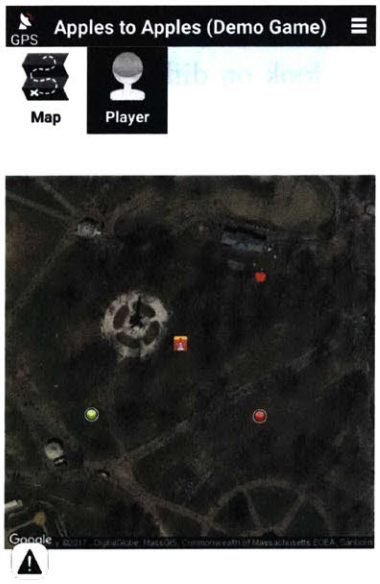
(a) No Background in City Environment



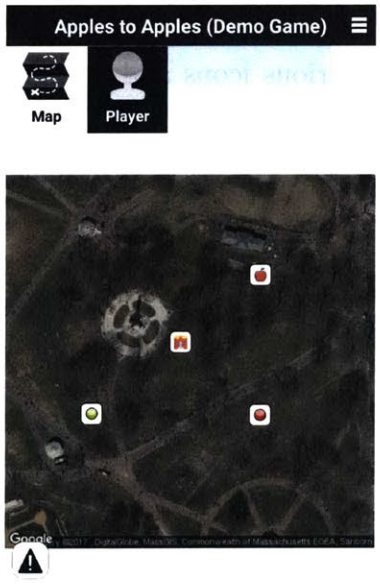
(b) White Rounded Rectangle in City Environment



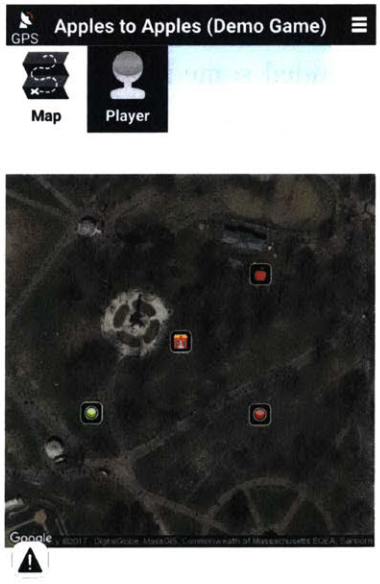
(c) Black Rounded Rectangle in City Environment



(d) No Background in Rural Environment



(e) White Rounded Rectangle in Rural Environment



(f) Black Rounded Rectangle in Rural Environment

Figure 4-7: Different Map Icon Backgrounds (None, White Rounded Rectangle, and Black Rounded Rectangle) on different types of maps (City and Rural)

4.3 Customizable Action Button Icons

Incorporating a new customizable icon feature for the action buttons gave me a chance to modify the existing UI in the TaleBlazer Editor for the actions buttons.

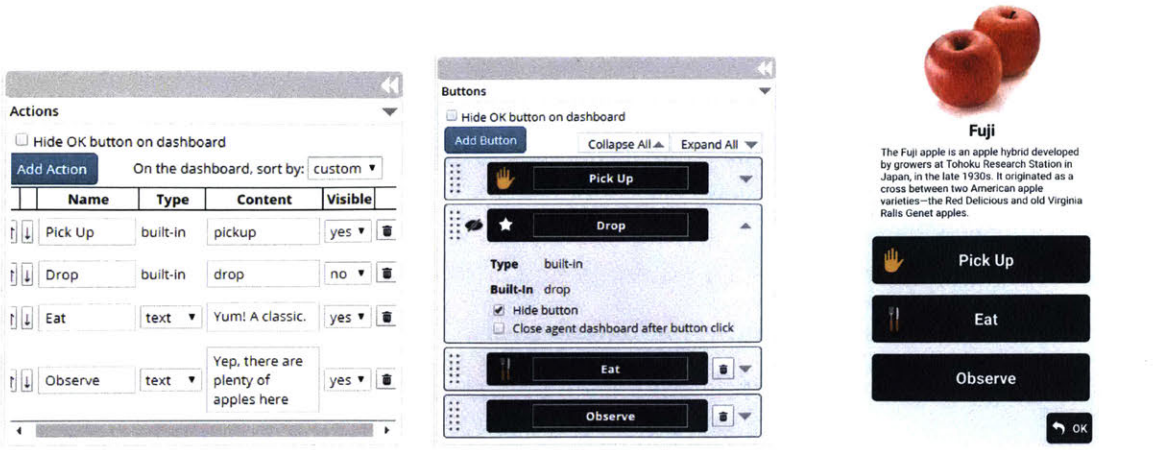
As mentioned in Section 4.1.1, there is now a diverse icon palette that is shared between the map icons and the action button icons. The palette can give more meaning to icons on the action buttons and scale on different screen sizes.

As seen in Section 3.2 (Figure 3-5), TaleBlazer action buttons were black with white text and a white icon. Now, since a designer can upload their own action button icons, the pattern can be preserved or changed. The user can choose to find white icons to place on the black button or they can incorporate colorful icons.

4.3.1 Action Table User Interface

Action button icons had a major problem. Since they were fixed in the mobile application, there was no way for the designer to preview how the agent buttons would look. In order to see the buttons, the designer had to try it on their own device or use the emulator. Thus, including a preview of the agent's action buttons is helpful for debugging and visualizing how the buttons will look.

In the previous version of TaleBlazer, the "Actions" table was very unintuitive and difficult to learn. Thus, the purpose of my redesign was to make the table easier to learn and understand. Many columns in the table had short headers that weren't immediately understandable. The columns names had to work across all buttons. In my redesign, I moved away from the table format so that the buttons could look unique based on their content and purpose.



(a) Old "Actions" Table (b) New "Buttons" Table (c) Representation on the Mobile (New Version)

Figure 4-8: The old version (left) vs. the new version (middle) of the Buttons Table in the Editor and the corresponding buttons on the Mobile (right)

Appearance and Name

In Figure 4-8, you can see that I renamed the section from "Actions" to "Buttons". Previously, game designers found the name "Actions" confusing and couldn't figure out how to add a button to an agent. Thus, the name "Buttons" in the Editor should help fix this problem. I also made each row look like a button in the mobile application. Thus, if a game designer is familiar with any TaleBlazer game, they should understand the parallels between this section in the editor and what appears in the mobile application.

Reordering Action Buttons

To reorder the actions in the old version, the game designer had to change the 'sort by' option from 'default' to 'custom' and use the arrows that appeared to move each row up or down (As shown in Figure 4-8a). This behavior was slow and cumbersome. If a designer had 10 buttons and wanted to make the first button last, they would have to click nine times to move the button to the desired position.

Thus, I implemented drag and drop in my redesign and removed the 'sort by' drop-down. This is faster, easier, and more intuitive.

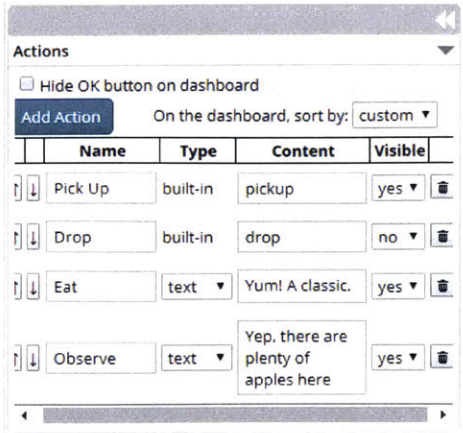


Figure 4-9: Old "Actions" Table in TaleBlazer Editor

Content Column

The content column was probably the most unclear column in the old version of TaleBlazer. As shown in Figure 4-9, the "Pick up" and "Drop" buttons have "pickup" and "drop" for their content respectively. The "Pick up" and "Drop" buttons are built-in buttons which every agent has. These buttons are present to support the inventory system that TaleBlazer utilizes. The content field for the "Pick up" and "Drop" buttons look editable, but they are not. As shown in Figure 4-8b, my redesign maps the word "Built-in" to "pickup" or "drop" and it doesn't look editable.

For buttons that aren't built-in, the "Content" text changes depending on the type of the button. As seen in Figure 4-10, for the "Video" type, the word "Video" maps to box for selecting the video that will play when that button is pressed. For the "Script" type, the word "Script" maps to box for selecting the script that will be executed when that button is pressed. And for the "Text" type, the word "Text" maps to box for typing the words that will display when that button is pressed.

Thus, the word "Content" is no longer floating around as a catchall descriptor.

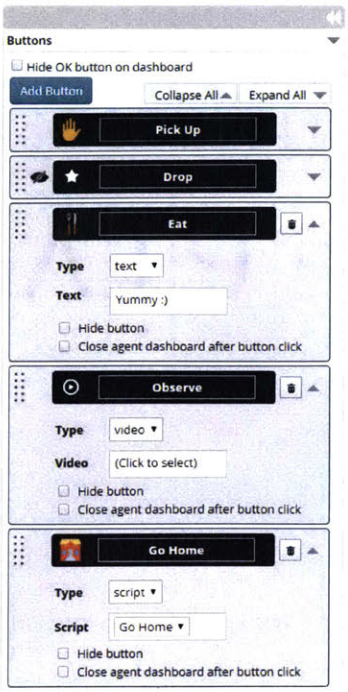


Figure 4-10: Different Types of Buttons in the Redesigned Editor

Visible Actions

Buttons can be hidden or visible in the mobile application. The game designer may decide to hide a button at the beginning of the game. Then, based on some choice the player makes, the button can change from hidden to visible (or vice versa). For example, imagine there is a "Market" agent which has the action "Buy Garment". The "Market" agent may only show the "Buy Garment" action if the player has enough money. If the player doesn't have enough money, the "Buy Garment" action is hidden.

In the previous version of TaleBlazer, the "Visible" column handled this behavior. As you can see in Figure 4-9, the "Visible" option is set to either 'yes' or 'no' in the old version of the TaleBlazer Editor. The wording of this column was not clear to novice game designers. Thus, in my redesign, I changed the look from a drop-down to a checkbox and added the label "Hide button" (Figure 4-10). Also, when the button is checked, the hidden icon appears next to the button (In Figure 4-10, the "Drop" button is hidden).

Every agent has a "Pick up" and "Drop" button that can not be deleted. By default, they are hidden for each agent.

Delete Button

As previously stated, every agent has a "Pick up" and "Drop" button. These buttons cannot be deleted. If a designer doesn't want to use these buttons, they can hide them (and by default, they are hidden for each agent). However, as seen in Figure 4-8a, these buttons had a delete button in the old version of the TaleBlazer editor. If a game designer clicks on the delete button, a pop-up appears with the text "You cannot delete built-in actions". Therefore, it doesn't make sense for these buttons to have a delete button in the first place. In my redesign, I removed the delete button for built-in actions and only included it for buttons you can actually delete (Figure 4-10).

Expand and Collapse

Previously, all the information for every button was visible all the time. It made everything easy to see, but a designer would also see a lot of information they didn't need to look at. In my redesign, I added "Expand All" and "Collapse All' buttons that show and hide all the button details (Figure 4-8b). If a designer wants to modify a specific button, they can click on the individual drop-down arrows on each button (In Figure 4-8b, the "Drop" button is expanded). This modification allows the list of buttons to be more compact at the click of a button, which makes it easier to see all the buttons at once.

Icon Preview

Buttons previously didn't have an option to change the associated icon. In the re-design, I created a new icon preview on the action buttons. The game designer simply clicks the icon and they are brought to the new screen (Figure 4-11) where they can change or update the icon. This screen is a variation of the Icon Picker for agent map icons (Figure 4-4 on page 39).

Designers may think that they cannot change the icon in the new version of the editor (since they couldn't change it previously). In order to make this feature more learnable, I added hover effects to the preview box in the redesigned editor. When a user hovers over the icon, the box changes color, the cursor changes from a pointer to a clicker, and a hint pops up that reads "Select action icon". The goal of these effects is to encourage people to understand that there is a new feature associated with action button icons.

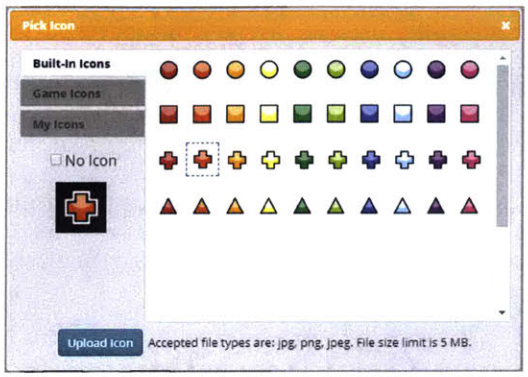


Figure 4-11: New Action Button Icon Picker in Editor

For action button icons, the game designer also has the choice to not use an icon on a button by checking the "No Icon" box in Figure 4-11.

Chapter 5

User Testing

After I created my features, I wanted to make sure that the features were learnable, intuitive, and usable. If the game designer can't find or learn my new features, the features are essentially useless. Thus, I included many UI hints that would help the designer identify the new features.

To ensure my new features were working an expected, I needed some user testers (also called users or testers) to test my designs (See Appendix C for the Full User Testing transcript).

5.1 User Testing Research Questions

Before interacting with users, I developed a list of 21 research questions about the new features to see if the features were intuitive and well thought out (Appendix C). I prioritized this list in order to hone in on my learning objectives. The questions were labeled high, middle, or low priority. High priority questions were seen as the most essential. If the users has trouble completing the tasks dealing with high priority questions, that means my features are hard to use or discover. Questions that weren't labeled were viewed as useful but nonessential.

I've included my high priority research questions below.

High Priority Research Questions [RQ]

Icon Picker Usability

1. Can the user select a built-in icon? [RQ1]
2. Can the user figure out how to upload an image? [RQ2]

Map Icon Usability

1. Can the user figure out how to change the map icon (ie open the icon picker)? [RQ6]
2. Can the user find and use the "Global Map Icon Settings"? Is it in an intuitive place? [RQ7]

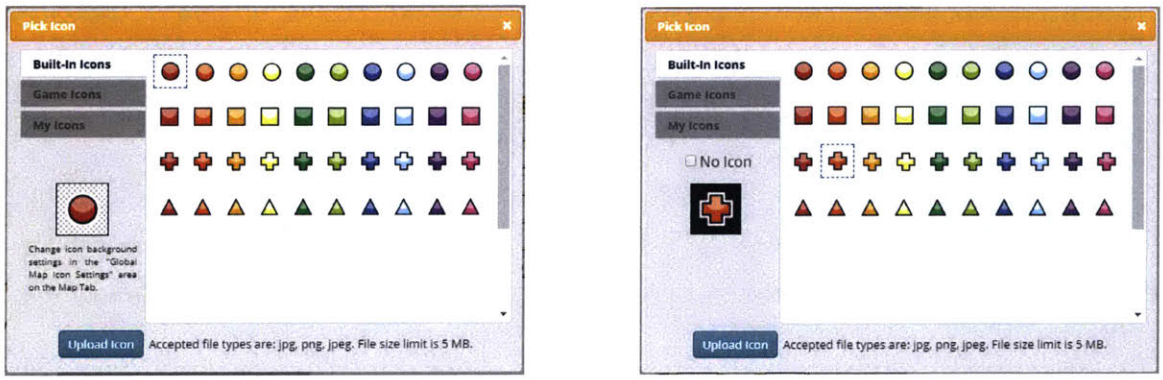
"Button" Section Usability

1. Can they figure out how to change the button icon (ie open the icon picker)? [RQ10]

These research questions shaped that tasks that I asked my testers to complete. I've explained the high priority questions in more detail below.

5.1.1 Icon Picker Usability

The two most important high priority questions had to deal with the map icon picker screen seen in Section 4-4 on page 39 and the action button icon picker screen seen in Section 4-11 on page 48 (These screens are replicated here for reference).



(a) New Map Icon Picker

(b) New Action Button Icon Picker

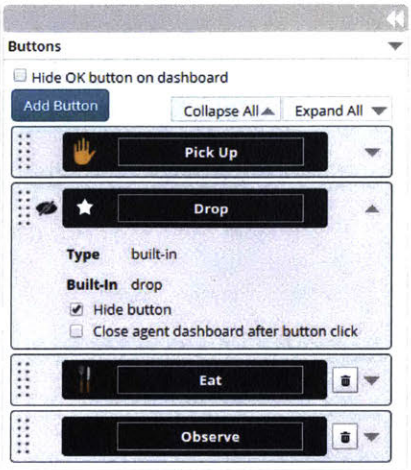
Figure 5-1: New Icon Pickers in the Editor

I wanted to ensure these screens were usable. Thus, the user should be able to select a built-in icon and upload their own image. The new font palette is included in the "Built-In Icons" section, but was excluded during user testing.

5.1.2 Icon Picker Discoverability

Additionally, the user needs to be able to find these icon picker screens by clicking either the map icon for an agent (to access the Map Icon Picker screen) or the button icon on a button (to access the Action Button Icon Picker screen). These two important areas are replicated here for reference and can be found in Section 4-5 on page 40 and Section 4-8 on page 44. If they can find the icon picker screens by clicking the correct area in the editor, they should be able to figure out how to change the icon.

(a) Agent Dashboard (to access Map Icon Picker)

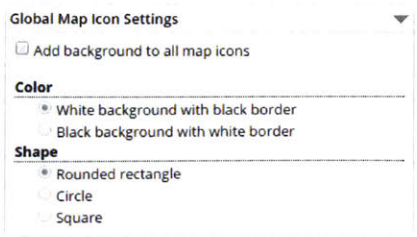


(b) New "Buttons" Table (to access Action Button Icon Picker)

Figure 5-2: Screens to access the new Icon Pickers in the Editor (All screens to access Map Icon Picker are shown in Figure 4-8 p. 44)

5.1.3 "Global Map Icon Settings" Usability

And finally, I wanted to see if they could locate and use the "Global Map Icon Settings" as seen in Section 4-3 on page 39 (Replicated here for reference).



(a) Map Icon Settings not used

Figure 5-3: New "Global Map Icon Settings" in Editor

5.2 User Testers

I interviewed two novice TaleBlazer users and two power TaleBlazer users during user testing. Each user was interviewed individually. Each interview took about 30 minutes. The two novice users were interviewed in person while I interviewed the two power users over Google Hangouts.

Novice users have little to no experience with TaleBlazer. They may have played TaleBlazer mobile games before, but they probably haven't used the editor extensively. Power users have familiarity with TaleBlazer and have created two or more games. They are very familiar with the TaleBlazer editor and the mobile application.

My two novice users were young children ages 11 and 12. My two power users were adults working in the industry. It was important to find novice and expert users, as well as interview different age groups, in order to see if the new features were compatible with all types of users.

5.3 User Testing Format

Based on my research questions, I developed tasks for the user to complete. I also created questions for the user to answer while they were competing each task in order to learn their option(s) about the new features. I judged each task on two criteria:

ability to complete the task and time spent completing the task. Thus, I wanted users to be able to complete the tasks dealing with my high priority question quickly. The users were not timed, but I noted when users were confused or stuck. The complete list of tasks and the corresponding questions asked are included in Appendix C.

5.3.1 Mobile Tasks

It is often easier to introduce new players to the mobile application side of TaleBlazer before the editor side. Thus, I gave my testers some mobile tasks on a demo game to get them familiar with TaleBlazer and the new features. During these tasks, I asked questions about the sizes of the map icons and the different options for the map icon backgrounds. I also asked about the new action button icons. The goals of these tasks were to see if users liked the new background feature for map icons, the new custom map icons, and the new look of the agent action buttons with icons.

5.3.2 Editor Tasks

After completing the mobile tasks, I showed the testers the editor side of TaleBlazer. Here, they completed tasks in the editor that produced the changes that they saw in the mobile tasks. Again, during these tasks, I asked questions about the editor. These questions were about the screen used to select or upload map or action button icons, the global map icon settings, and the new redesign of the "Buttons" table. These tasks were designed to see if the global map icons settings or the icon selector screen were learnable and discoverable. I also wanted to see if the new redesigned "Buttons" section was usable.

Example Editor Task: Icon Picker Usability

Here are two examples of editor tasks that I asked the users to perform. Subtask 1 addresses high priority research questions 1 and 6 (RQ1 and RQ6). The user should be able to find the map icon picker and change the map icon. Subtask 3 addresses high priority research question 2 (RQ2). The user should be able to upload their own

image for a map icon. All research questions are listed in Appendix C and a list of high priority questions can be found in section 5.1.

Map Icon Instructions

Subtask 1 [RQ1, RQ3, RQ6, RQ18]

1. Create a New Game in the Editor
2. Go to the "Agents" Tab
3. Describe "Agent 1"'s map icon
4. Change "Agent 1"'s map icon to a different "Built-in" Icon

Questions

- Could you easily find the icon picker widget?

5. Go back to the icon picker widget

Questions

- Do you like the placement of the preview box? Is anything confusing?

Subtask 3 [RQ2]

1. Upload "dog.png" from the Desktop to represent "Agent 1" on the map

Questions

- Could you easily figure out how to upload an image for the map icon?

Table 5.1: User Testing Editor Map Icon Subtasks 1 and 3

Each task is aimed to address certain research questions. During each task, the user is guided to the corresponding part of the user interface and asked to perform a task and answer questions. If the user is unable to perform the task or answer a question, then something about the user interface may be wrong. User mistakes are valuable because they may indicate that something about the design of the user interface is confusing or unclear.

5.4 Results

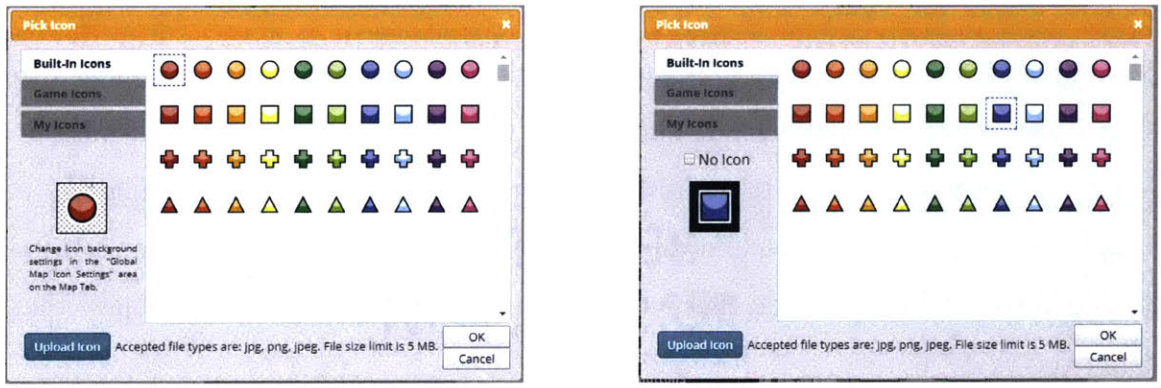
After user testing, I realized that I needed to make some slight adjustments. Overall, I was happy with the results and how well the testers were able to use my new features. I received positive results for 4 out of 5 of my high priority questions. The users were able to select a built-in icon (RQ1), upload their own images (RQ2), and change the map icons (RQ6) and button icons (RQ10). Half of the users struggled with finding the global map icon settings (RQ7), so more work can potentially be done in that area to make the future more discoverable.

Mobile Tasks

During the mobile tasks, all users said they liked the size of the map icons and that the icons were visible. This is positive feedback, however all interviews were conducted inside. Thus, opinions could differ if the interviews were conducted outside[1]. All user testers said the rounded rectangle background option was their favorite. This is important because this option is the first one listed on the Map Preview page in the TaleBlazer editor. Thus, it looks like I made the right choice making the rounded rectangle the first shape option (As seen in Figure 5-3).

Editor: Icon Picker

The two novice users had trouble confirming their choices for icons. They both accidentally pressed the "Upload Icon" button instead of exiting the screen by clicking the "X". Therefore, I decided to include "Okay" and "Cancel" buttons on the Icon picker. That way users will know there is a button for uploading and separate buttons for confirming or cancelling.



(a) Icon Picker for Map Icons

(b) Icon Picker for Action Button Icons

Figure 5-4: Icon Picker with Ok/Cancel Buttons

Editor: "Buttons" Section

One power user thought the action button icons should have a light outline around them (similar to the light outline around the box for the name of the buttons) so that

[1]My interviews could not be conducted outside because of bad weather and distance. My power users and myself were in different time zones, while my novice users were interviewed on a rainy day.

they would look more editable.

Editor: Global Map Icon Settings

There were some things I knew the testers would want which the testers confirmed while they completed the tasks. They expected the map preview on the map tab to update when they selected a background for the map icons. They also wanted the ability to have different map icon settings for each individual agent. I address both of these concerns in the following section (See Chapter 6 on page 57). The power users were both able to figure out where the global map icon settings were in the editor. They both read the hint on the Icon Picker screen, which led them to the Map Tab. Both of the novice users needed help finding the global map icon settings. This implies that this feature may be easy to miss for some users.

Novice vs. Power Users

I also noticed that there was a knowledge gap due to age. The two novice users didn't know what the hidden symbol on the action button tiles meant. However, the two power users were familiar with the icon and intuitively knew what it meant. The hidden icon is used in other applications, such as Photoshop, when dealing with hiding and showing layers. Thus, adult users who have seen the hidden symbol in other applications are going to understand the meaning of the symbol. Similarly, the novice users thought the drag and drop symbol (the 2 columns of dotted lines on the left side of each action button tile) was decoration, while the power users knew that it implied that the tiles were draggable.

Chapter 6

Future Work

While working on this project, there were a few features that I didn't get to fully develop due to time restrictions. There are also many completely new customizable features that can be implemented in the future. These features would give the game designer more freedom to create theme in their games while also making the games easier to play and understand.

6.1 Customizable Agent Map Icons

A huge missing component in the editor is the absence of a WYSIWYG[1] view of agent icons on the map tab when the global map icon settings are used (Figure 6-1). This means that game designers can't see how the map icons will look on the map in the editor with the backgrounds included.

Also, users would likely want the ability to change the global map icon settings to individual map icon settings. That way, the game designer can choose if each region has a different background or if certain agents have a background. However, mixing background shapes and colors could potentially make the game look less cohesive. Thus, there are pros and cons to allowing the game designer to individualize the map icon settings.

[1] What You See Is What You Get

Figure 6-1: "Map Tab in the Editor with no Map Icon Settings used" and "Map Tab in the Editor with Map Icon Settings set to white circle background with black border" look exactly the same

And finally, if the number of icons in the default palette grows, future work in TaleBlazer could focus on sectioning off icons based on themes such as nature, sports, music, technology, etc. This would make it easier for game designers to find the icon they are looking for (or discover if the icon they need is not in the icon palette). Tagging the icons so that they are searchable and including a "search" feature would also be a nice improvement.

6.2 Customizable Action Button Icons

In the future, TaleBlazer could allow game designers to change the color of the buttons. Since a font is included in the icon palette, game designers may want to change the color of the font so that the icon color changes.

6.3 Customizable Tab Icons

Under the "Settings" Tab in the TaleBlazer Editor, designers can configure the tabs that appear on the mobile version of the game. Game designers can customize tab names, but the tab images are preselected. Thus, inconsistencies between the image

and the tab name can occur. As shown in Figure 6-2, clicking the "Configure Tabs" button brings up the menu shown in Figure 6-3. Here, game designers can select which mobile tabs will appear in the game. They can also change the names of these tabs. As shown in Figure 6-2, it is also possible to switch the order of the tabs. One simply drags and drops the tabs into the desired order.

Figure 6-2: Mobile Tabs Screen in the TaleBlazer Editor that allows the game designer to switch the order of the tabs, by dragging and dropping

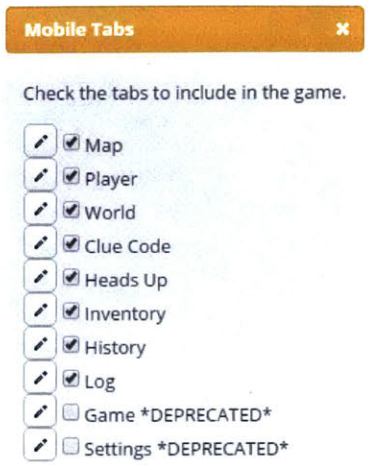


Figure 6-3: Mobile Tabs Screen that allows the game designer to select which tabs are shown and gives them the option to rename those tabs

Once these tabs are set, they have fixed icons in the mobile game. Figures 6-4 and 6-5 show the fixed icons that are displayed for each of the options shown in Figure 6-3.

Since designers can change the names of these tabs, it would be helpful if they could customize their own images as well. This way, designers could avoid disconnects between the tab name and the tab image. For example, if the game has a space theme and the designer names the "Map" tab "Space", it's a little confusing for the tab to have the preselected treasure map image (Figure 6-4). This would also further the uniqueness and theme of each game. Creating a default palette for tab icons would also be helpful, but would be more challenging, since designers can change the tab name to anything.

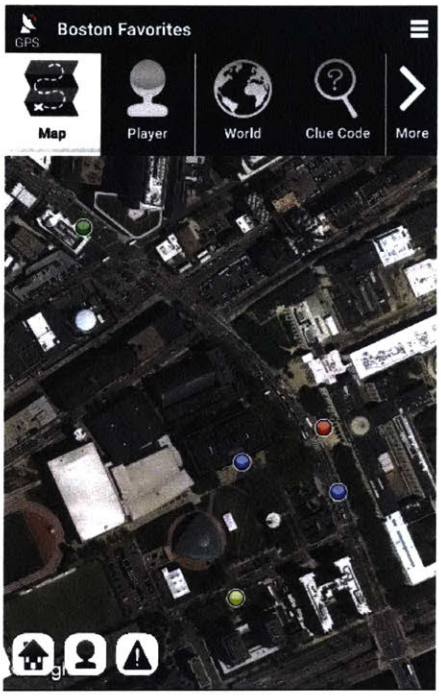


Figure 6-4: Fixed Icons for Map, Player, World and Clue Code tabs

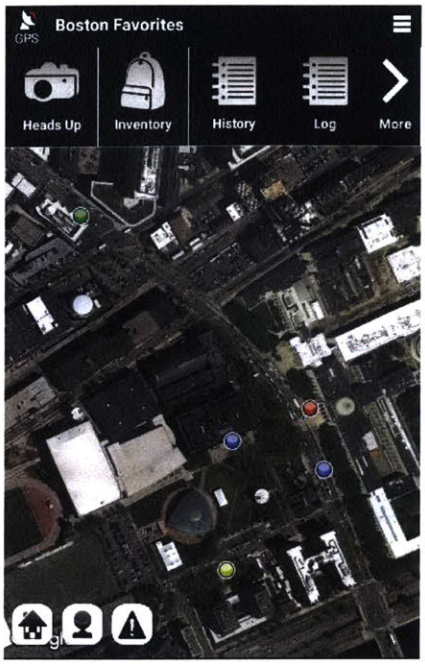


Figure 6-5: Fixed Icons for Heads Up, Inventory, History, and Log tabs

6.3.1 Challenges with Customizable Tab Icons

Customizable tab icons are a challenge because there is not a clear set of defaults that can work for most applications. Each of the tabs has a specific meaning and the designer can change the name of the tab, but the meaning is the same. Thus, tab icons contribute more to the theme of a game. Since TaleBlazer can't determine the designer's desired theme, there is no immediately obvious pool of default images. Thus, creating a descriptive palette of images may prove to be difficult.

Also, all the tab icons look a certain way when they are selected. As seen in Figure 6-4, selected tabs are white with a black icon and black text, while deselected tabs are black with a white icon and text. Therefore, if a designer can upload their own tab images, this pattern should be preserved.

However, using a large font for the icon palette potentially solves the two previously mentioned problems of image diversity and pattern preservation. As stated

before, font icons can easily change color and the particular font I found has over 1,500 images that easily rescale to different sizes. Thus, my font icon palette may be usable for customizable tab icons in the future.

Lastly, the TaleBlazer Editor does not currently provide a way to view how the tab icons will look in the mobile game. In order to test the customizable tab icons, the designer will have to try it on their own device. This is not really conducive to using this feature. It would be favorable to have a WYSIWYG tab icon preview.

6.4 Customizable Agent Sound Effects

Currently, within TaleBlazer, there are sounds when you bump an agent and when you enter a correct or incorrect clue code (clue codes are text strings in TaleBlazer that can be used to unlock hidden agents). These sounds are fixed, but it would be valuable if game designers could change or suppress the agent sounds. For example, a happy effect could play when the player bumps a helpful or positive agent. And similarly, a malicious effect could play if the player bumps a harmful or negative agent. TaleBlazer could provide some default sound effects and allow the game designer to use their own.

Customizable agent sound effects could also lead to the incorporation of helpful invisible agents. For example, if an invisible agent is bumped, it could play a clip that indicates that the player is proceeding in the correct direction. Or the sound effect for the invisible agent could notify the player that there is something nearby.

6.4.1 Challenges with Customizable Agent Sound Effects

Customizable agent sound effects are difficult because providing a group of sound effects that can be used as defaults may be hard to collect. Also, if the designer wants to upload their own files, we might need to make decisions about the maximum length of a sound effect.

There is also the challenge of figuring out if a sound effect should be played just the first time it is bumped or every time it is bumped. If the effect is played once, the player could lose the hint or information, but if it is played every time, that could be a hindrance to the gameplay. It may be beneficial to let the game designer make this decision with some kind of toggle.

Finally, including sound effects with agents will be challenging because the existing framework for customizable icons (images) doesn't aid with the development of customizable sound effects (sounds). Thus, the work that I've done on icons can't be used to work with sounds.

6.5 Viewing Gifs

Game designers can upload custom agent images to their games, but the gif file extension does not behave as expected in the mobile application. You can add gifs to objects like roles, agents, inventory items and the world dashboard. When you add images to these TaleBlazer objects, the screen in Figure 6-6 is shown. Here, the instructions say that the "Accepted file types are: jpg, png, jpeg, gif". In Figure 6-6, an example of a gif is selected as an agent icon. In the editor, the image is static. Also, when the same agent is viewed within the TaleBlazer mobile game, as shown in Figure 6-7, the image remains static. Despite such language in the TaleBlazer Editor that might lead the game designer to expect full support for gifs in TaleBlazer, gif files are displayed as static images and not as short animations in the Editor and on the mobile.

Thus, it would be advantageous to add full gif support to TaleBlazer such that gifs are displayed properly as short silent animations in the editor and in the mobile app. Gifs are powerful because they can be used to show instructions to players or add dimension to objects and characters.

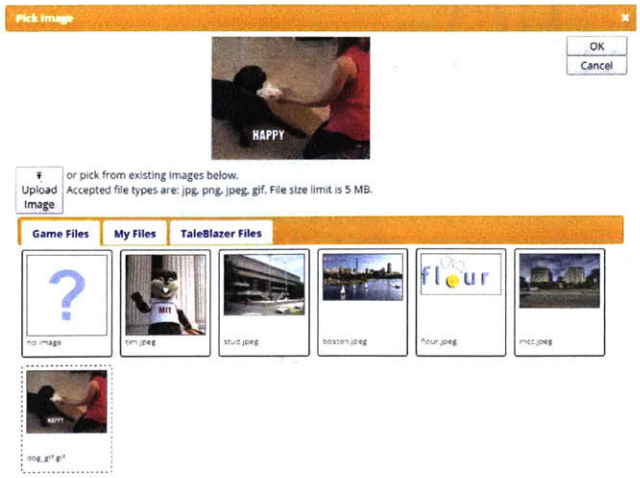


Figure 6-6: The "dog_gif.gif" file is added to an agent in the TaleBlazer Editor. A static image is shown to the designer within the editor.

Figure 6-7: The "Dog Gif" Agent in the TaleBlazer mobile application shows a static image just

6.5.1 Challenges with Viewing Gifs

It would be beneficial to enable displaying gifs versus static images in the mobile application and in the editor. An option could be to manually show gifs by displaying each individual image of the gif on a timer (so that it looks like the continuous movement in the original gif). The mobile side doesn't currently support gifs, so that would need to be changed. Currently, gifs are compressed into static images and stored on the server, so all existing gif files on the server would never be able to be animated.

6.6 Other Customizable Tasks

There are a number of other customizable tasks that I brainstormed (allowing the game designer to customize the clue code page, adding visuals to roles, customizing fonts and colors etc.), but I focused on Customizable Agent Map Icons and Customizable Action Button Icons. I believed that these were the most meaningful changes. Also, my work in these areas should make future customizable tasks (like customizable tab icons) easier.

Chapter 7

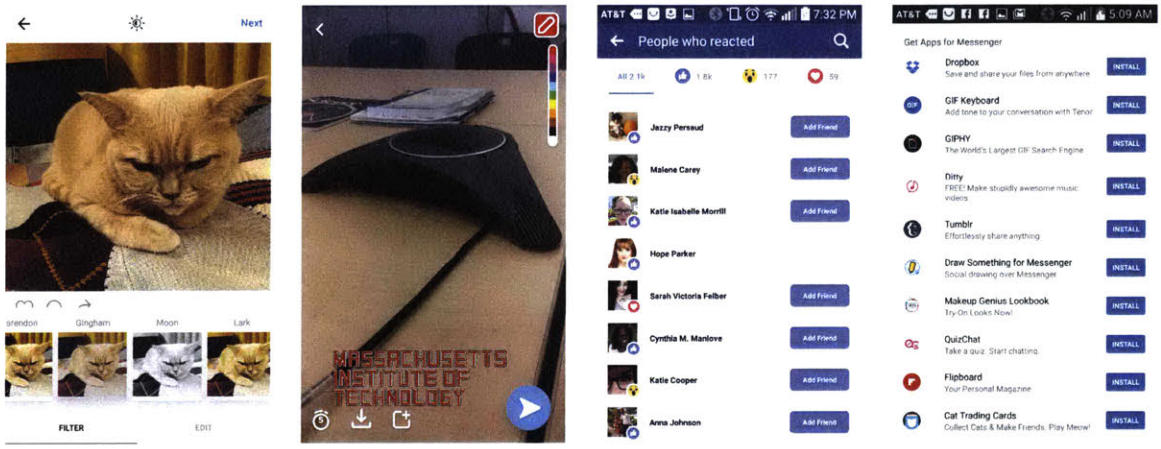
Conclusion

My goal was to include more customizable features for game creators in TaleBlazer. Customization adds meaning and ease of use to applications. Thus, incorporating new customizable features will give game designers the ability to create games with appearances that harmonize with the goal of the game. It can also make games more visually interesting and unique for each developer. This, in turn, should make the game easier for users to play and understand.

My new features, customizable map icons and customizable agent action button icons, give game designers greater variety and freedom while creating their TaleBlazer games. These features are both easy to use and intuitive. They were also user tested across different age groups and expertise levels. Future work in the area of customization will continue to enhance gameplay in TaleBlazer games and create more positive user experiences.

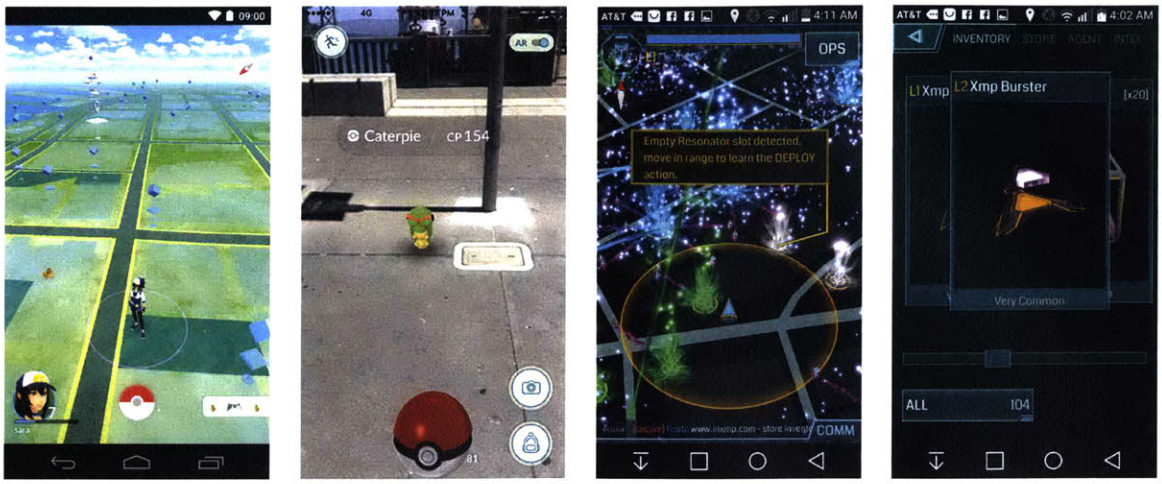
Appendix A

Figures



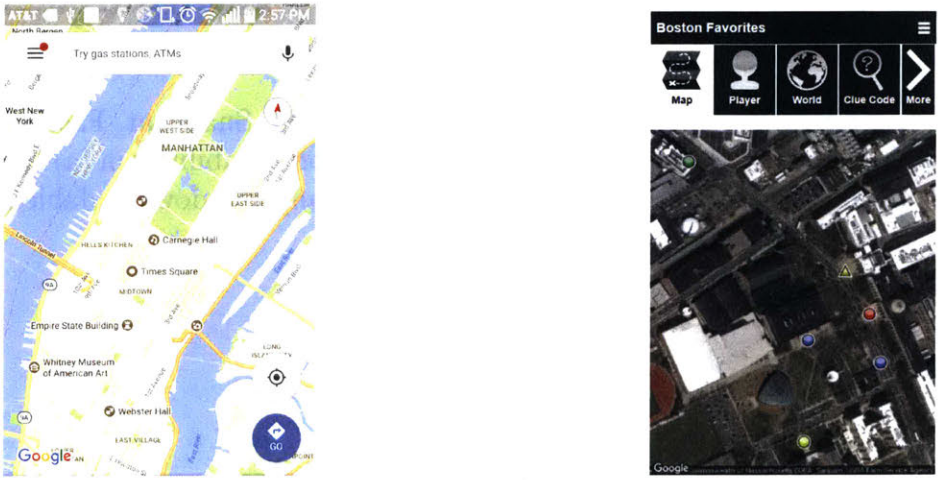
(a) Instagram (b) Snapchat (c) Facebook (d) Messenger

Figure A-1: Screenshots of (from left to right) Instagram, Snapchat, Facebook, and Facebook Messenger illustrating the image customization capabilities in (a) Instagram and (b) Snapchat, a reactions page on (c) Facebook, and the downloadable add-ons available for customizing (d) Facebook Messenger.



(a) Pokémon Go Map (b) Pokémon Go Gameplay (c) Ingress Map (d) Ingress Gameplay

Figure A-2: Screenshots of the(from left to right) Pokémon Go Map, Pokémon Go Pokémon Capture Screen, Ingress Map, Ingress Inventory screen



(a) Google Maps (b) TaleBlazer

Figure A-3: Google Maps (Left) and TaleBlazer map screen (Right) on mobile devices

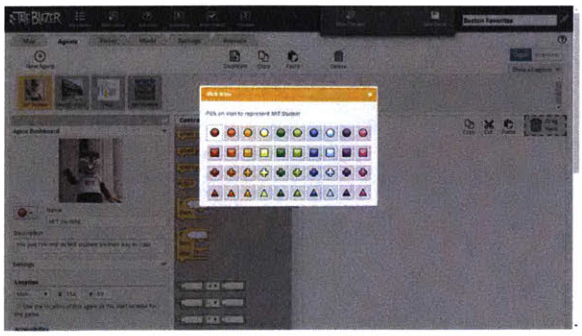


Figure A-4: The Agent Tab in the Old TaleBlazer Editor (background) with the "Pick Icon" Screen (foreground)

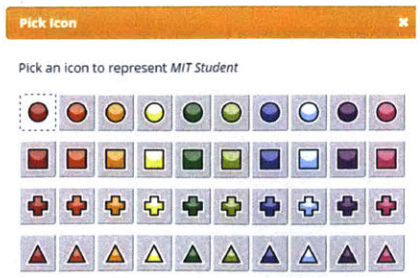


Figure A-5: The "Pick Icon" Screen in the TaleBlazer Editor (Old Version)

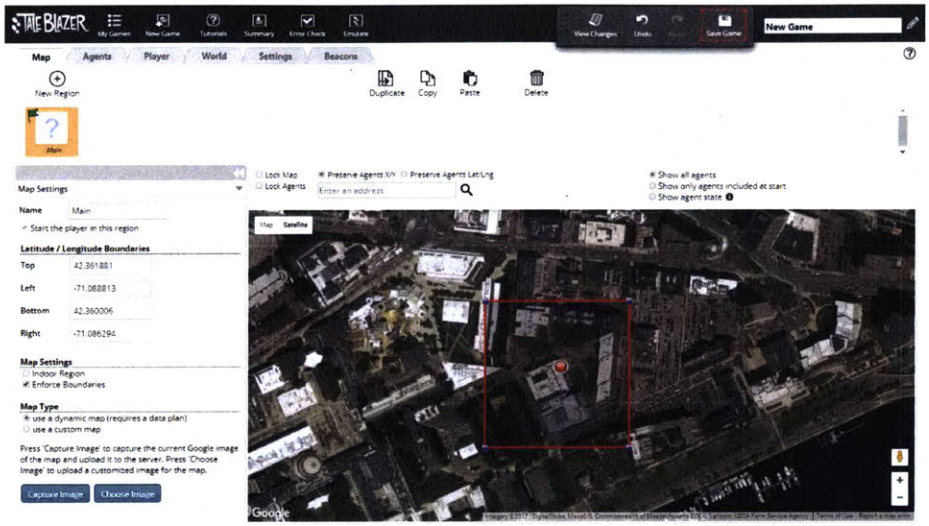


Figure A-6: The Map Tab in the TaleBlazer Editor (Old Version) with the red circle representing an agent

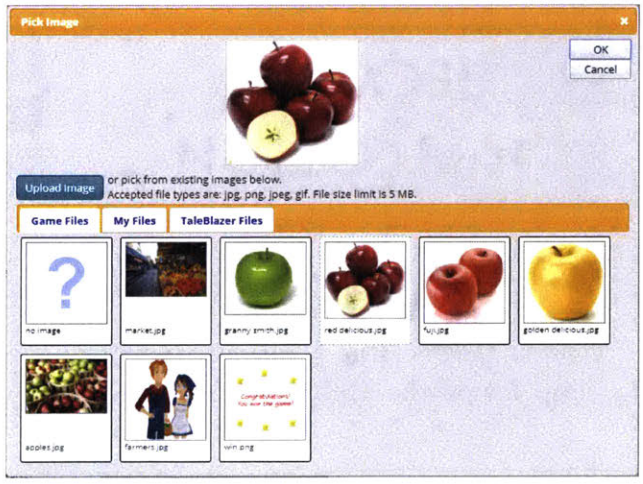
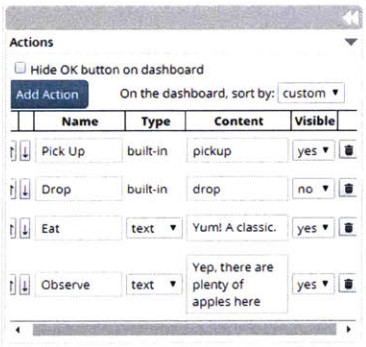


Figure A-7: An example of the Image Picker in TaleBlazer Editor with some user uploaded images

Figure A-8: Action Buttons on TaleBlazer Mobile (Old Version)

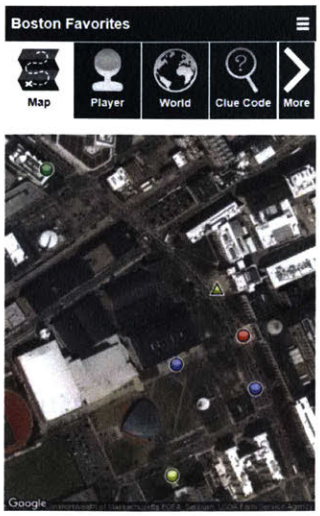


(a) "Actions" Table in TaleBlazer Editor (Old Version)

(b) Corresponding buttons on TaleBlazer Mobile (Old Version

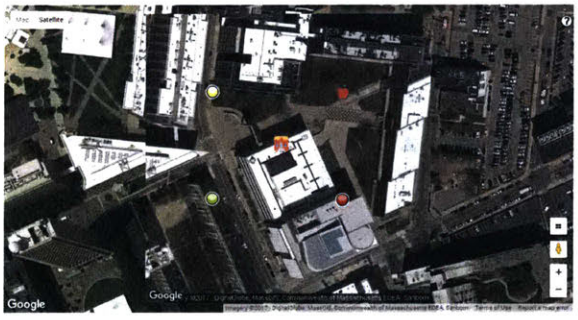
Figure A-9: The old version (left) of the "Actions" Table in the TaleBlazer Editor and the corresponding buttons on the Mobile (right)

(a) Old Version of "Map" Tab in TaleBlazer Editor



(b) Corresponding "Map" Tab in Mobile app

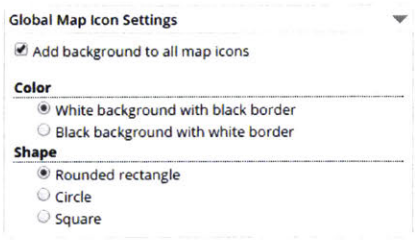
Figure A-10: Map preview on the "Map" Tab in the old version of TaleBlazer (left) and the corresponding "Map" Tab on the mobile application (right)



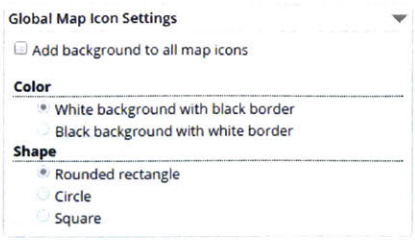
(a) New Version of "Map" Tab in TaleBlazer Editor

(b) Corresponding "Map" Tab in Mobile app

Figure A-11: Map preview on the "Map" Tab in the revised version of TaleBlazer (left) and the corresponding "Map" Tab on the mobile application (right)



(a) Map Icon Settings with a white rounded rectangle background and black border



(b) Map Icon Settings not used

Figure A-12: New "Global Map Icon Settings" in Editor

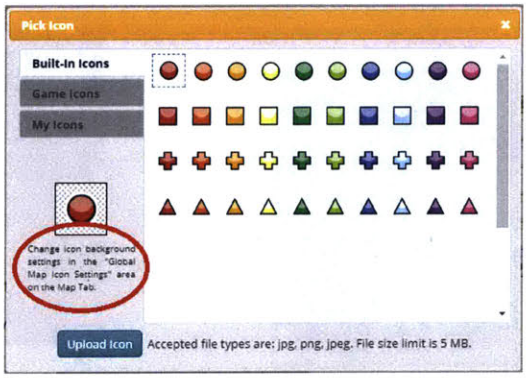
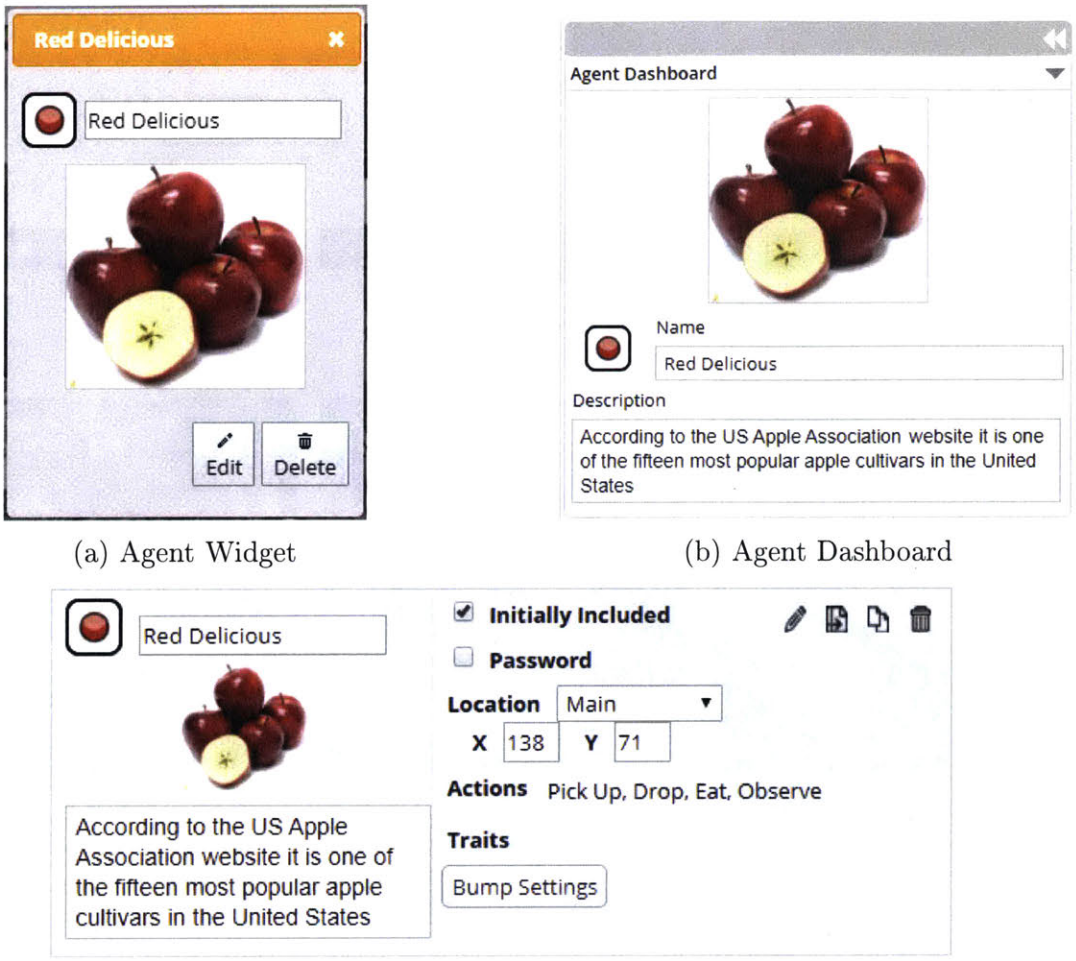


Figure A-13: Help Dialog to help game designers find the new feature

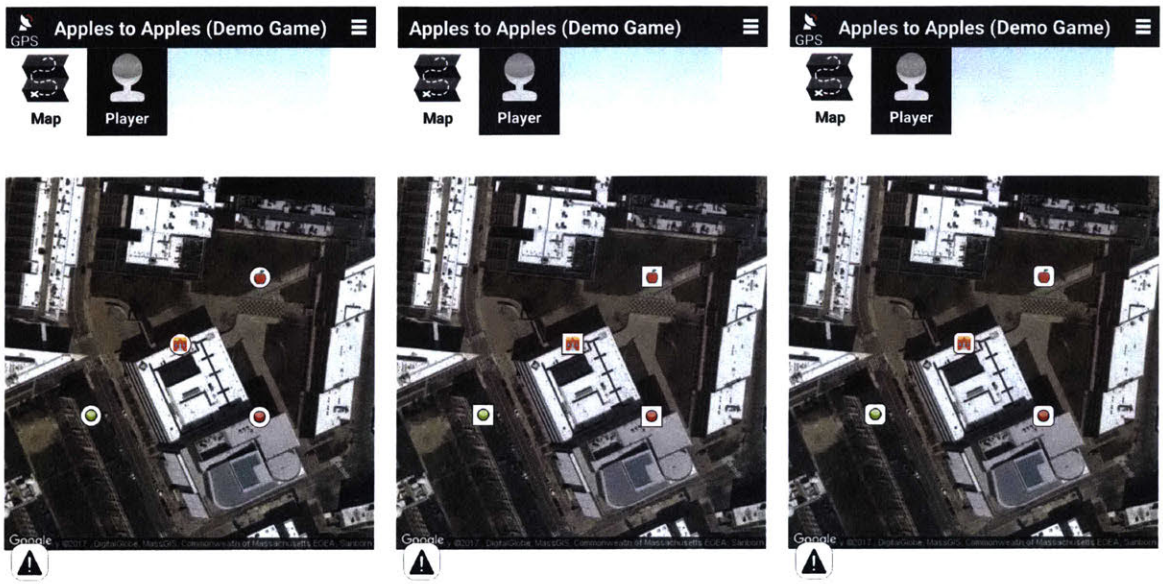


(a) Agent Widget

(b) Agent Dashboard

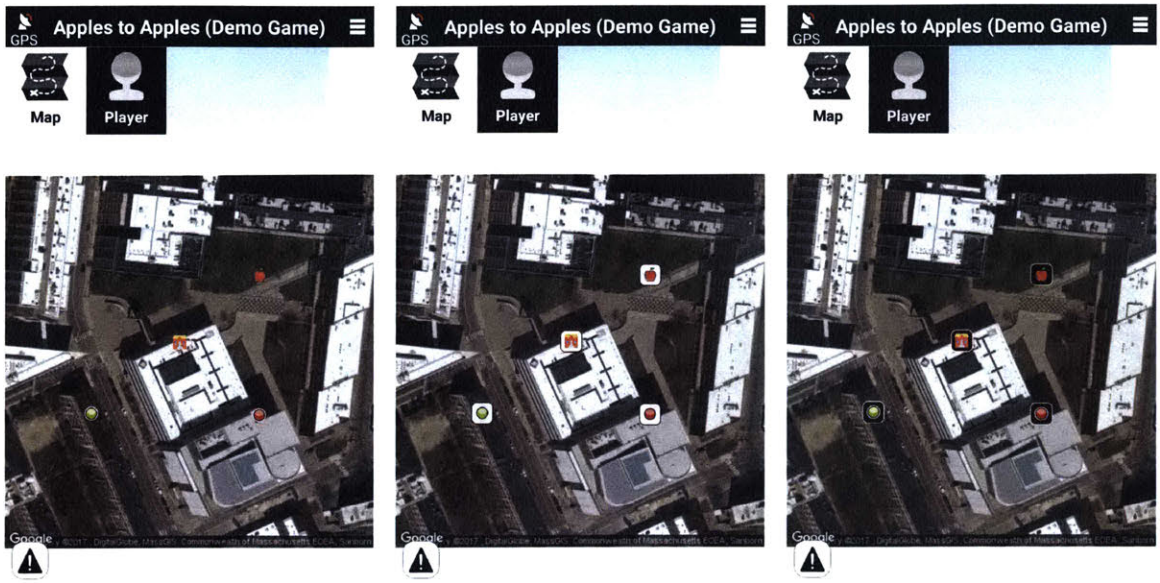
(c) Agent Overview

Figure A-14: Changes in the New TaleBlazer Editor when the map icon settings are set to white rounded rectangle background with black border in the Agent Widget (Top Left), Agent Dashboard (Top Right) and Agent Overview (Bottom)



(a) Circle (b) Square (c) Rounded Rectangle

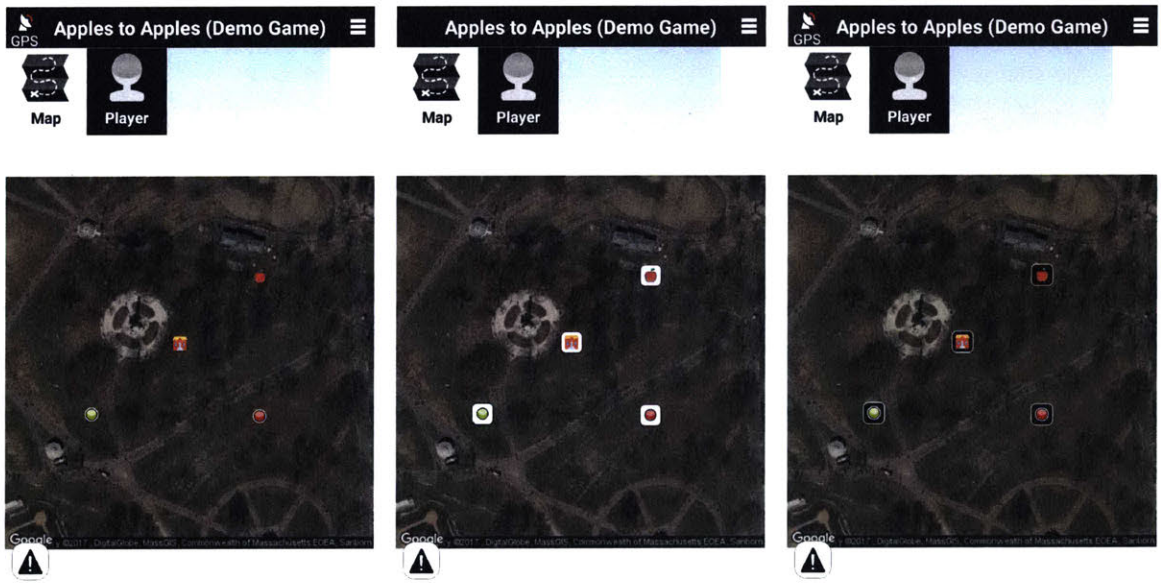
Figure A-15: Images of the same mobile TaleBlazer game with a white background and black border. Shape set to Circle (left), Square (center) and Rounded Rectangle (right)



(a) No Background in City Environment

(b) White Rounded Rectangle in City Environment

(c) Black Rounded Rectangle in City Environment

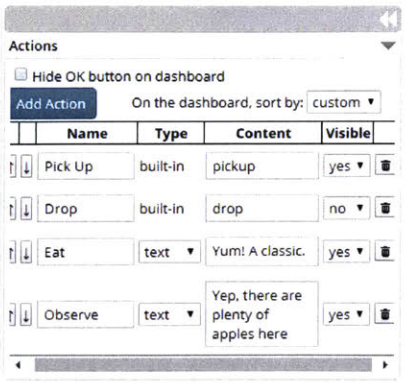


(d) No Background in Rural Environment

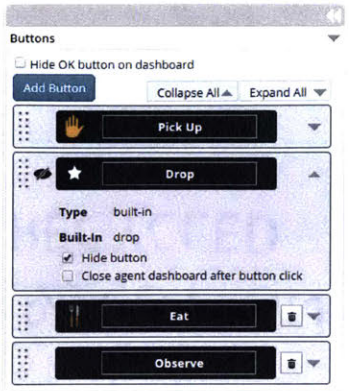
(e) White Rounded Rectangle in Rural Environment

(f) Black Rounded Rectangle in Rural Environment

Figure A-16: Different Map Icon Backgrounds (None, White Rounded Rectangle, and Black Rounded Rectangle) on different types of maps (City and Rural)



(a) Old "Actions" Table



(b) New "Buttons" Table

(c) Representation on the Mobile (New Version)

Figure A-17: The old version (left) vs. the new version (middle) of the Buttons Table in the Editor and the corresponding buttons on the Mobile (right)

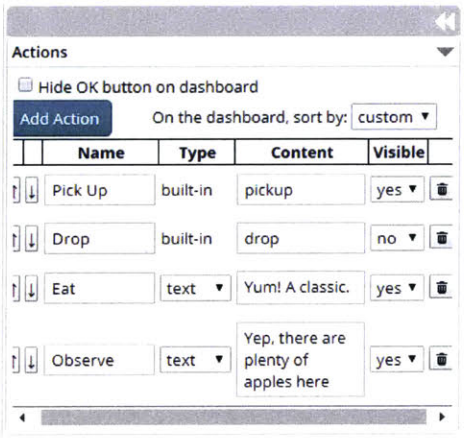


Figure A-18: Old "Actions" Table in TaleBlazer Editor

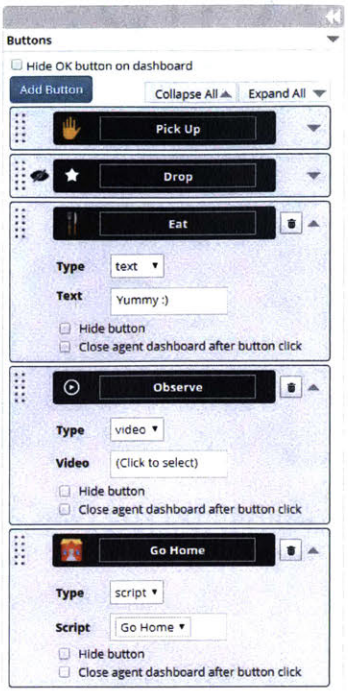


Figure A-19: Different Types of Buttons in the Redesigned Editor

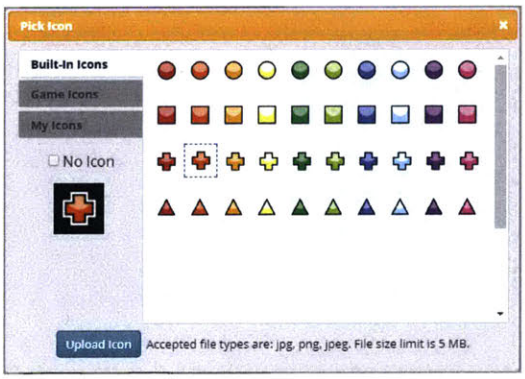


Figure A-20: New Action Button Icon Picker in Editor

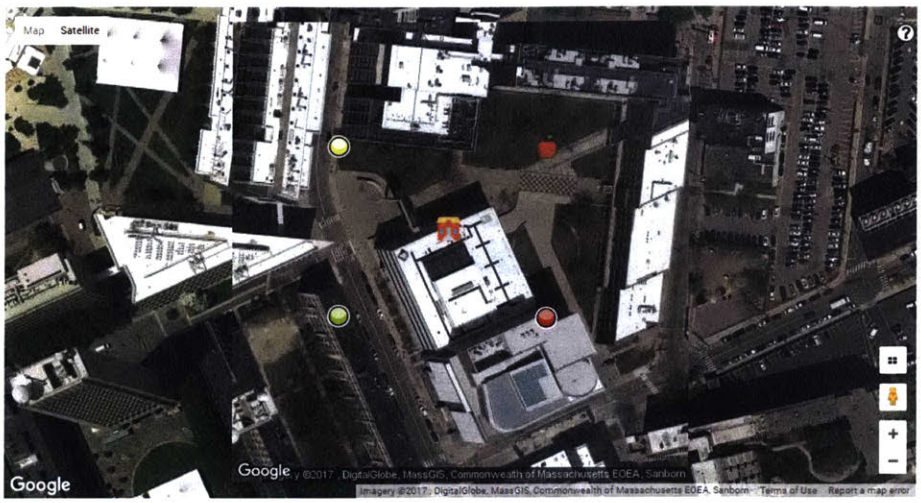


Figure A-21: "Map Tab in the Editor with no Map Icon Settings used" and "Map Tab in the Editor with Map Icon Settings set to white circle background with black border" look exactly the same

Figure A-22: Mobile Tabs Screen in the TaleBlazer Editor that allows the game designer to switch the order of the tabs, by dragging and dropping

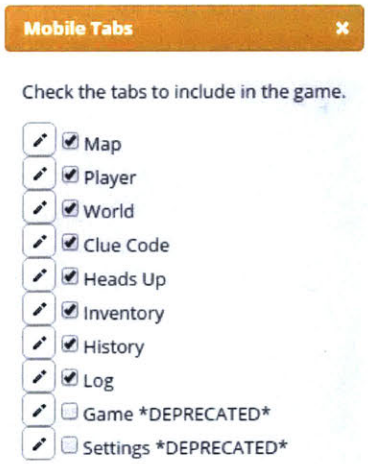


Figure A-23: Mobile Tabs Screen that allows the game designer to select which tabs are shown and gives them the option to rename those tabs

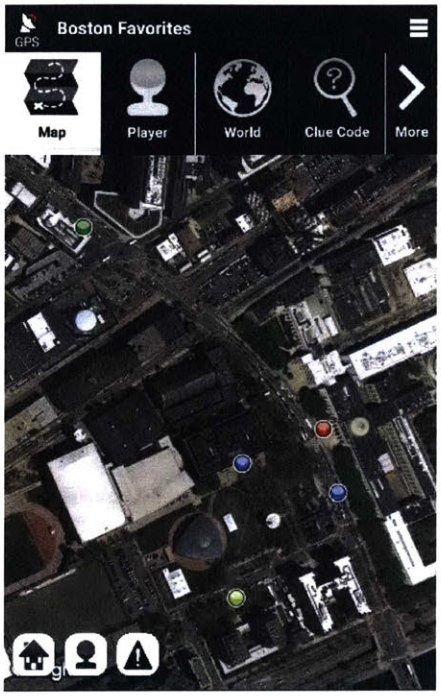


Figure A-24: Fixed Icons for Map, Player, World and Clue Code tabs

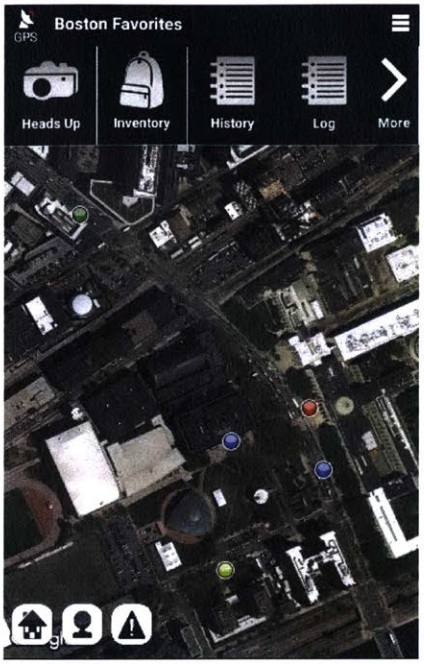


Figure A-25: Fixed Icons for Heads Up, Inventory, History, and Log tabs

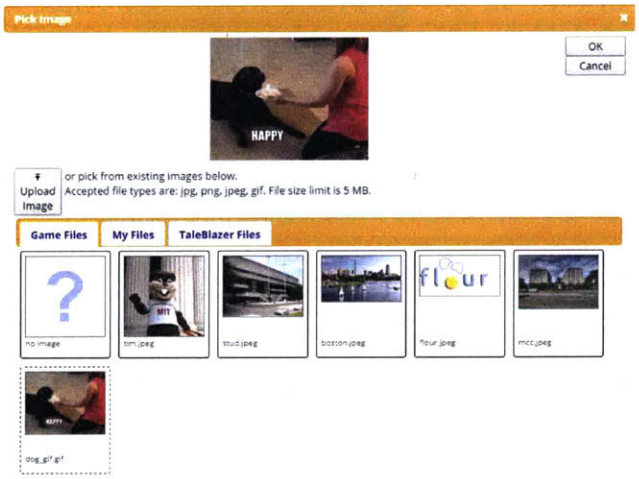
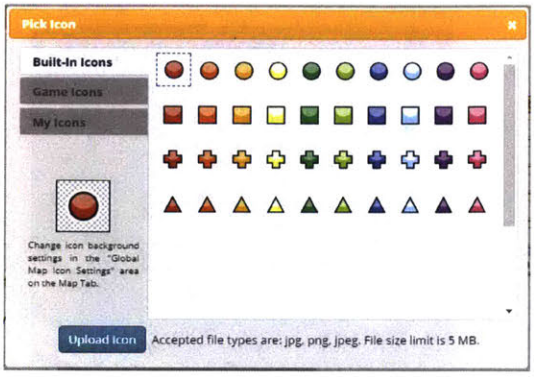


Figure A-26: The "dog_gif.gif" file is added to an agent in the TaleBlazer Editor. A static image is shown to the designer within the editor.

Figure A-27: The "Dog Gif" Agent in the TaleBlazer mobile application shows a static image just

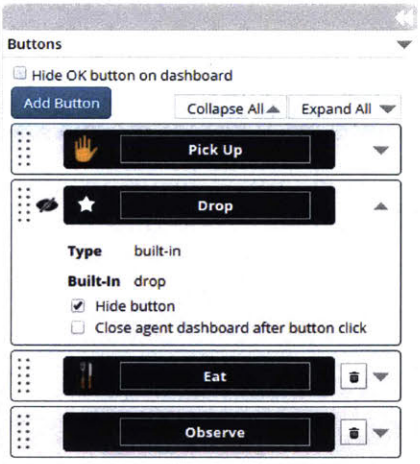


(a) New Map Icon Picker

(b) New Action Button Icon Picker

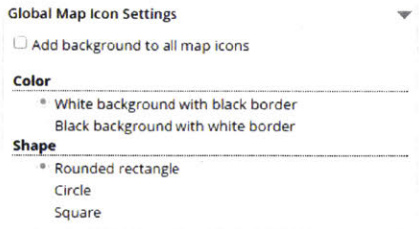
Figure A-28: New Icon Pickers in the Editor

(a) Agent Dashboard (to access Map Icon Picker)



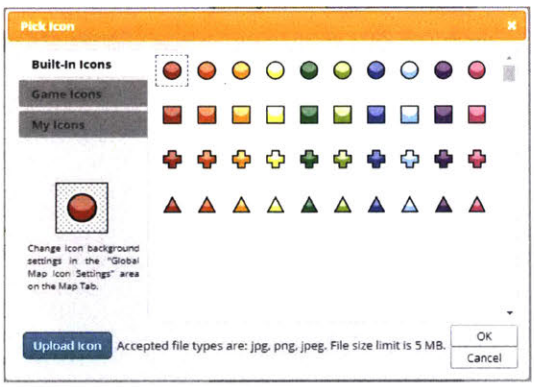
(b) New "Buttons" Table (to access Action Button Icon Picker)

Figure A-29: Screens to access the new Icon Pickers in the Editor (All screens to access Map Icon Picker are shown in Figure 4-8 p. 44)

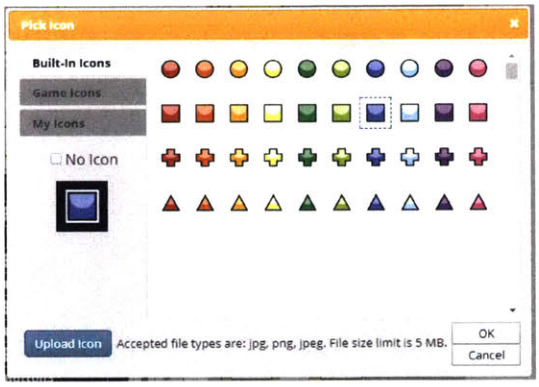


(a) Map Icon Settings not used

Figure A-30: New "Global Map Icon Settings" in Editor



(a) Icon Picker for Map Icons



(b) Icon Picker for Action Button Icons

Figure A-31: Icon Picker with Ok/Cancel Buttons

Appendix B

Tables

Criteria	Font(s)	Images
Diverse set of Images	Need to find the smallest number of fonts that gives the greatest diversity	Find all individual images
Adjustable size for different screen resolutions	True	Possible Pixilation
Easy to Expand	Needs to be in the font already or find new font	Find new images
Adjustable Color	True	False
Effect on Mobile App Size	Depends on font(s) size(s)	Depends on images sizes and number of images
Monochromatic[1]	True	True or False

The Criteria for the Icon Palette

[1]Multicolor icons look good in most applications, but might be hard to see in mobile TaleBlazer games outside. Thus, a monochromatic palette is more favorable.

Font(s)	
Pros	Cons
Adjustable color and size	Entire font must be included in the mobile application (increases the size of the application)
Adjustable size without pixilation	Can be repetitive (contain variations of the same symbol)
	Can contain unusable/irrelevant symbols (like symbols for decoration)
	Monochromatic
	Limited to the icons within the font (need to add new fonts if you want to include more icons)

The Pros and Cons of using Fonts for the Icon Palette

Images	
Pros	Cons
Monochromatic or Colorful	Need to pick a good size that won't be pixelated or too small on different screen resolutions
Can include more images without rebuilding the mobile application	Have to find each individual image

The Pros and Cons of using Images for the Icon Palette

Criteria	Font(s)	WebHostingHub Glyphs
Diverse	Need to find the smallest number of fonts that gives the greatest diversity	1,514 glyphs
Adjustable size for different screen resolutions	True	True
Easy to Expand	Needs to be in the font already or find new font	Shouldn't need to expand
Adjustable Color	True	True
Effect on Mobile App Size	Depends on font(s) size(s)	One large font
Monochromatic[2]	True	True

The WebHostingHub Glyphs Font meets most of the criteria

[2]Multicolor icons look good in most applications, but might be hard to see in mobile TaleBlazer games outside. Thus, a monochromatic palette is more favorable.

Appendix C

TaleBlazer "Icon" Testing Guide for Testers

This appendix gives the guide used by the proctor of the user testing. User testers saw a shortened version of this guide that just included the tasks. User testers did not see the questions or background information.

C.1 Description of Icon Feature

The new icon feature allows game designers to select from a wider palette of icons than the previous version of TaleBlazer. If the palette does not meet the game designer's needs, they can use the editor to upload their own custom icons to agents and actions. Icons can enrich the quality of gameplay if utilized properly, thus, having a variety of icons is important for the game designer, as it allows them to convey new information to the game player. Furthermore, the option to upload custom icons provides a way for the game designer to include the icons they want.

Map Icons

For map icons, the game designer has the option to adjust the global map icon settings. These optional icon settings affect how the agent icon will appear on the map during gameplay by supplying a background color, shape, and border for the enclosed icon.

There are two different types of map icons:

- Built-in icons: The icons that are built into the game. Previously, the game included 40 icons for agents (4 shapes and 10 colors)
- Custom icons: The icons that the game designer uploads into the game.

Action Button Icons

For action icons, the game designer has the option to select a blank icon as the icon for an action. This option is useful in case the game designer does not want to use an image at all on a particular action.

There are three different types of action icons:

- Built-in icons: The icons that are built into the game. Previously, the game included 3 icons for actions (the speech bubble for text, the play buttons for video and the star for scripts).
- Custom icons: The icons that the game designer uploads into the game.

- Blank icons: The game designer may select that an action button have no icon displayed. This option is only available for action buttons.

We would like to test the changes to both the editor and mobile app, and gauge the usability and functionality of the new feature. There are two (or three) tasks we've designed for user testers to complete:

1. Play a demo game to test mobile changes
2. Complete a few tasks on the new version of the editor

Details for each of these tasks are listed below. Each task includes a series of questions.

C.2 Research Questions

[HIGH] High Priority [MID] Middle Priority [LOW] Low Priority

Icon Picker Usability

1. Can the user select a built-in icon? [HIGH]
2. Can the user figure out how to upload an image? [HIGH]
3. Do they understand the preview box?
4. Do they understand the difference between built-in icons and game icons?
5. Do they want to use the same icon set for the agent icons and the button icons?

Map Icon Usability

6. Can the user figure out how to change the map icon (ie open the icon picker)? [HIGH]
7. Can the user find and use the "Global Map Icon Settings"? Is it in an intuitive place? [HIGH]

"Button" Section Usability

8. Can they figure out how to change the button icon (ie open the icon picker)? [HIGH]
9. Do users see the parallel between the "buttons" section and the mobile?
10. Is the "Buttons" section easy to understand? [MID]
11. Can the user figure out how to set an agent's action icon to a blank icon? [MID]
12. Can the user tell which buttons are hidden? [MID]
13. Do users know they can drag and drop the elements in the table (intuitive)? [LOW]
14. Should the action tile be dragged by only the left side or by the whole row? [LOW]
15. Should the default state of the icons be closed or open? [LOW]

Button Icon Functionality

16. Are there too many options or not enough options for the button icon? Do you understand the images?

Map Icon Functionality

17. Are the options for setting the map icon background (rrect, circle, square) (white w/black border vs black w/white border) overwhelming, inadequate, or about right? [LOW]
18. Are the options for setting the agent's map icon overwhelming, inadequate, or about right?[LOW]

Mobile

19. What is the optimal appearance of the visited custom icons (with and without the background)? [LOW]
20. What is the optimal appearance of the visited built-in icons (with and without the background)? [LOW]
21. What is the optimal size for the map icons (with and without the background)? [LOW]

Future questions

- Do the users want to add tags to icons?
- Should buttons without icons be centered with respect to the button or aligned with the words of buttons with icons?

C.3 Researcher Script

Hello, my name is Victoria. I am a graduate student at MIT working on adding new features to a game platform called TaleBlazer. TaleBlazer allows people to make and play their own location-based augmented reality games. If you've played PokÃĺmon Go, you've played a location-based augmented reality game.

I'm interviewing people like you to get feedback on the new features I've been working on. I want your honest opinion so that I can make the software even better. I'm going to show you the software and ask you to do a few tasks. I would like you to talk aloud as you go through the tasks and let me know about any problems or questions you have. If something is confusing to you, it will probably be confusing to someone else and it would be very helpful to me to understand that. I am not testing you, I am testing the software, so feel free to say anything and everything that comes to mind.

Intro to Task 1

TaleBlazer games are meant to be played outside. The player will be told to 'walk to the icons to visit them'. In this test, we will tap on the icons instead to visit the characters in the game.

Intro to Task 2

Subtasks 1-3

The TaleBlazer editor is used by designers to make new TaleBlazer games. In these tests, we will explore the editor to create new things within the game.

After Subtasks 4 and 5

For map icons, the game designer has the option to adjust the global map icon settings. These optional icon settings affect how the agent icon will appear on the map during gameplay by supplying a background color, shape, and border for the enclosed icon. The game designer can also see these changes reflected in several places in the editor.

Task 1: Play a Demo Game

Instructions

Please remember to read the instructions aloud and vocalize anything you're thinking. It helps me understand what is confusing. Whenever you see Questions, please pause and I will ask you some questions about what you see.

Subtask 1

1. Select the demo game "Apples to Apples" on Device 1
2. Look at the map

   Questions:

   - Are the icons large enough for you to see them clearly on the map?
   - Do you think the icons are too big, too small, or just right?
   - Do the icons cover too much of the map?
3. Select/Visit an agent

   Questions:

   - Are the icons large enough for you to tap them easily?

Subtask 2

1. Look at the map of the demo game on Devices 2, 3 and 4

   Questions:

   - Do you think the icons with the background are too big, too small, or just right?
   - Do the icons with the background cover too much of the map?
   - How do you like the thickness of the border of the backgrounds?
   - Which background is your favorite?

Subtask 3

1. Visit the agent that looks like an apple on Devices 1 and 2
2. Look at the agent's buttons

   Questions:

   - Do you think the button icons are too big, too small, or just right?
   - Do you like the look of the button with no icon?
3. Click 'Ok' to go back to the map

   Questions:

   - Do you like the look of the visited custom icon on Device 1 and 2? Do you have a favorite?

4. Visit the agent on the screen that looks like a red circle in Device 1 and Device 2
5. Click 'Ok' to go back to the map

Questions:

- Do you like the look of the visited built-in icons on Device 1 and Device 2? Do you have favorite?

Subtask 4

1. Play through the demo game on Device 1

— End of Task 1 —

Task 2: Work with the Editor

Map Icon Instructions

Subtask 1 [RQ1, RQ3, RQ6, RQ18]

1. Create a New Game in the Editor
2. Go to the "Agents" Tab
3. Describe "Agent 1"'s map icon
4. Change "Agent 1"'s map icon to a different "Built-in" Icon

Questions

- Could you easily find the icon picker widget?

5. Go back to the icon picker widget

Questions

- Do you like the placement of the preview box? Is anything confusing?

Subtask 2 [RQ4]

1. Go to the "Agents" Tab
2. Find the dialog to change the map icon
3. Explore all the tabs

Questions

- Do you like how these tabs look?
- What do you think the different tabs mean?

Subtask 3 [RQ2]

1. Upload "dog.png" from the Desktop to represent "Agent 1" on the map

Questions

- Could you easily figure out how to upload an image for the map icon?

—— Now we will do some tasks relating to the global map icon settings. ——

Subtask 4 [RQ7]

1. Create 2 more agents
2. In the editor, figure out how to add a white square background to all of the map icons

Questions

- Were the settings where you expected them to be? Why or Why not?
- Did the settings make sense?
- If anything, what do you expect to happen?

Subtask 5 [RQ17]

1. Find the global map icon settings
2. Look at all the options

Questions

- Do you like the different options for color and shape?
- Do you think there should be more or less options?

—— Now we will do some tasks relating to the action icons. ——

Task 2: Work with the Editor

Action Button Instructions

Subtask 1 [RQ10]

1. Go to the "Agents" Tab
2. Find the area where you would add buttons to an agent

Questions

- What do you think the down arrow on each tile means?

Subtask 2 [RQ10]

1. Expand all the tiles at once
2. Collapse all the tiles at once
3. Expand just the "Drop" button

Questions

- Do the "Expand All", "Collapse All", and the individual down arrow behave as expected? Why or why not?

Subtask 3 [RQ10]

1. Add a new button to the agent
2. Expand the "Button1" button
3. Change the type from "Text" to "Video"
4. Change the button name from "Button1" to "New Button"
5. Delete the "New Button" button

Questions

- Was anything confusing or different than you expected?

Subtask 4 [RQ8, RQ11, RQ12]

1. Add a new button to the table
2. Set "Button1" so that it does not show any icon on the button
3. Set the icon for the "Drop" button to the icon that looks like a play button
4. Set the "Pick up" to be visible.

Questions

- Is it useful to have the "No Icon" feature? [Experienced User]
- Is the placement of the "No Icon" checkbox where you expected it to be?
- Can you see the white icons against the grey background?
- Explain what the eye icon means. Did you know what the eye icon meant before this task?

General Questions [RQ9, RQ10, RQ13]

- Did you like the look of the Button section? Why or why not?
- Is there anything you would change?
- Is anything unclear about the Button section?
- Did you know you can drag the button tiles?
- Do you think the buttons look similar to the mobile buttons? Do you like this?

General Editor Questions

- Was anything about your experience with the editor interface unintuitive?
- Were you confused or frustrated at any point, even slightly? Please describe/explain where/why.

— End of Task 2 —

Bibliography

[1] TaleBlazer, http://taleblazer.org/, Accessed Oct 2016

[2] The MIT Education Arcade and Scheller Teacher Education Program "About" Section, http://education.mit.edu/about/, Accessed Oct 2016

[3] Quartz "These are the 25 most popular mobile applications in America", https://qz.com/481245/these-are-the-25-most-popular-2015-mobile-apps-in-america/ , Accessed Oct 2016

[4] ScriptBlocks, https://code.google.com/p/scriptblocks/, Accessed Oct 2016

[5] Snapchat, https://www.snapchat.com/, Accessed Oct 2016

[6] Instagram, https://www.instagram.com/, Accessed Oct 2016

[7] Facebook, https://www.facebook.com/, Accessed Oct 2016

[8] Facebook Messenger, https://www.messenger.com/, Accessed Oct 2016

[9] Ingress, https://www.ingress.com/, Accessed Oct 2016

[10] Pokémon Go, http://www.pokemon.com/us/pokemon-video-games/pokemon-go/ , http://pokemongo.nianticlabs.com/en/, Accessed Oct 2016

[11] WebHostingHub Glyphs, https://www.fontsquirrel.com/fonts/webhostinghub-glyphs, Accessed July 2017